THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

délivré par **l'Institut Supérieur de l'Aéronautique et de l'Espace**

Spécialité : Intelligence Artificielle

Présentée et soutenue par **Sylvain Bouveret**

Le 16 novembre 2007

**Allocation et partage équitables de ressources indivisibles :
modélisation, complexité et algorithmique**

JURY

M. Christian Bessière, président du jury
M. Ulle Endriss
M. Thibault Gajdos
M. Jean-Michel Lachiver, co-directeur de thèse
M. Jérôme Lang, co-directeur de thèse
M. Michel Lemaître, co-directeur de thèse
M. Patrice Perny, rapporteur,
M. Thomas Schiex

École doctorale	:	**mathématiques, informatique et télécommunications de Toulouse**
Unité de recherche	:	**équipe d'accueil SUPAERO-ONERA MOIS (ONERA-DCSD, centre de Toulouse) - IRIT - CNES**
Rapporteurs	:	**M. Boi Faltings et M. Patrice Perny**
Directeurs de thèse	:	**M. Jean-Michel Lachiver, M. Jérôme Lang et M. Michel Lemaître**

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce travail de thèse. Ils ont le droit à ma reconnaissance et à un exemplaire gratuit de ce manuscrit s'ils le souhaitent. Je tiens toutefois à adresser quelques remerciements particuliers aux personnes qui seront citées dans les deux pages suivantes, et qui m'ont très certainement apporté autant sur le plan technique que sur le plan humain.

En premier lieu, je tire un coup de chapeau à Jean-Michel Lachiver, Jérôme Lang et Michel Lemaître, qui m'ont encadré pendant ces trois années. Ils ont su m'apporter, par leur relative disponibilité, leurs différences, et leurs qualités techniques et humaines, tous les ingrédients nécessaires à la réalisation d'un travail de thèse dans des conditions excellentes. J'espère avoir encore moult occasions de les côtoyer, que ce soit pour une discussion technique, à l'autre bout du monde, ou encore au détour d'un chemin ou d'une falaise.

Simon de Givry, Thomas Schiex et Gérard Verfaillie ont eu l'occasion de suivre mon travail de près à l'occasion en particulier de leur participation à mon comité de thèse, même si leur influence sur ce travail dépasse largement ce cadre restreint. Je les en remercie vivement ! Merci particulièrement à Thomas Schiex d'avoir accepté de faire partie de mon jury.

Merci infiniment aux Professeurs Boi Faltings et Patrice Perny, qui ont eu la lourde tâche probablement quelque peu fastidieuse d'être les rapporteurs du présent manuscrit. J'ai eu plaisir à discuter avec eux, à l'occasion de multiples rencontres, et leurs conseils m'ont été très utiles pour améliorer ce manuscrit ainsi que pour la suite de mon travail. J'aimerais remercier en outre Patrice Perny pour m'avoir accueilli dans son équipe au Laboratoire d'Informatique de Paris 6 pendant deux semaines.

Je remercie Christian Bessière, Ulle Endriss et Thibault Gajdos, membres de mon éclectique jury de thèse, pour la qualité de leur analyse, la précision de leurs questions et la pertinence de leurs conseils lors de la soutenance. Merci à Christian Bessière d'avoir accepté de présider le jury. Merci à Thibault Gajdos d'avoir accepté de s'aventurer sur les terres inconnues de l'informatique en participant à mon jury de thèse. Je remercie enfin chaleureusement Ulle Endriss, pour avoir lu le manuscrit avec autant d'attention, pour toutes les discussions que nous avons pu avoir au cours de ma thèse, pour l'énergie qu'il déploie à organiser des événements fédérateurs dans la communauté, et pour m'avoir donné l'occasion de participer à de nombreuses reprises à des groupes de travail.

J'ai effectué ma thèse au sein de l'équipe Conduite et Décision du Département Commande des Systèmes et Dynamique du vol de l'Onera, où j'ai bénéficié de conditions de travail particulièrement agréables. J'aimerais donc remercier Patrick Fabiani et Jean-François Gabard de m'avoir accueilli dans cette équipe, ainsi que tous les membres de l'unité dont j'ai apprécié la compagnie. Merci à Christophe "Tof" Garion, de Supaéro, de m'avoir fait confiance et de m'avoir fait découvrir le monde merveilleux de Java et de l'enseignement. Merci aussi à tous les personnels (techniques, administratifs, gardes, etc.) qui nous facilitent la tâche au quotidien et contribuent à nous rendre le travail plus agréable.

D'un point de vue plus pragmatique, je remercie le Cnes et l'Onera d'avoir financé cette thèse. Je remercie aussi l'ANR pour le financement des missions dont j'ai pu bénéficier à plusieurs reprises au cours de ma thèse, dans le cadre de ma participation au projet Phac.

Outre les personnes que j'ai déjà citées dans les paragraphes précédents, j'ai eu la chance de faire de multiples connaissances durant ma thèse, au détour de groupes de travail, de conférences ou de séminaires. Je pense tout d'abord à toutes les personnes qui gravitent autour du groupe MultiAgent Resource Allocation et du projet ANR Phac, et en particulier les gens du Lamsade dont Yann Chevaleyre, Sylvia Estivie et Nicolas Maudet, du Cril, de l'ILLC, et du Lip6. Je remercie à cette occasion particulièrement l'équipe Décision du Lip6, qui m'a accueilli chaleureusement pendant deux semaines, et notamment (outre Patrice Perny que j'ai déjà cité), Olivier Spanjaard, Paul Weng, et la joyeuse équipe des thésards, Lucie, Nicolas, et les autres. Merci aussi à Bruno Zanuttini pour toutes nos discussions communes, pour ses qualités humaines, et pour son invitation à l'Université de Cæn.

Merci à la meute des thésards toulousains, parmi lesquels :

- les thésards du DCSD et assimilés : Clément et ses sandales, Cédric et son professeur émérite K. Hungus, l'autre Cédric, Nico, Greg le millionnaire, Greg le prolétaire, Manu tentionnaire, Charles, Olivier, Flo, Stéphane, Julien, Patrice, Sophie, Florent et les autres ;
- les piliers du DMAE : Claire et sa polaire frottée, Bruno, inventeur du cadeau concept et de la loi de Frackowiak, Nico ;
- la horde de l'Irit : Élise, Kévin, Nico ;
- les joyeux drilles de l'Inra : Matthias, dont l'activité principale consiste à compiler le C ;
- les autres : rajoutez votre nom ici si je vous ai oublié : ....

Merci à Barth de m'avoir hébergé tant de fois lorsque j'étais en mission. Merci aux membres de la confrérie Saint–Luc, issus d'une longue lignée de maîtres brasseurs depuis 2007. Merci aux amis proches et moins proches, aux grimpeurs, aux nageurs, aux cinéphiles, aux cynophiles, aux amateurs de bovins, aux skieurs, aux randonneurs, aux chercheurs, aux étudiants, aux coureurs, aux piliers de comptoir, aux adeptes du jeu de mots, aux *geeks*, aux mélomanes, aux ours des montagnes,...

Merci à tous les professeurs qui m'ont donné l'envie d'apprendre, la curiosité intellectuelle, et le goût de la recherche.

Merci à ma mère de m'avoir soutenu et donné l'opportunité de faire des études.

Merci enfin à Marianne, pour tout le reste, tout ce qui compte réellement.

Je dédie cette thèse à Alice, née le 26 janvier 2006.

Table des matières

Notations

Ensembles

$\mathbb{R}$, $\mathbb{Q}$ et $\mathbb{N}$	Ensemble des réels, des rationnels et des entiers naturels
$\overline{\mathbb{R}}$, $\overline{\mathbb{Q}}$ et $\overline{\mathbb{N}}$	$\mathbb{R} \cup \{-\infty, +\infty\}$, $\mathbb{Q} \cup \{-\infty, +\infty\}$ et $\mathbb{N} \cup \{+\infty\}$
$\mathbb{R}^+$ et $\mathbb{Q}^+$	Ensemble des réels positifs et des rationnels positifs
$\mathbb{R}_\star$, $\mathbb{Q}_\star$ et $\mathbb{N}_\star$	$\mathbb{R} \setminus \{0\}$, $\mathbb{Q} \setminus \{0\}$ et $\mathbb{N} \setminus \{0\}$
$\mathbb{R}^+_\star$ et $\mathbb{Q}^+_\star$	$\mathbb{R}^+ \setminus \{0\}$ et $\mathbb{Q}^+ \setminus \{0\}$
Lettre calligraphique	Ensemble mathématique ($\mathscr{X}$, $\mathscr{Y}$,...)
$\{\ldots\}$	Ensemble mathématique
$\times$	Produit cartésien
$\mathscr{X}^n$	$\underbrace{\mathscr{X} \times \cdots \times \mathscr{X}}_{n \text{ fois}}$
$\wp(\mathscr{X})$	Ensemble des parties de $\mathscr{X}$
$\subset$, $\supset$	Relation d'inclusion
$\subseteq$, $\subsetneq$	Inclusion au sens large, inclusion stricte
$\in$	Appartenance à un ensemble
$[a, b]$	Intervalle réel fermé entre a et b
$[\![n, m]\!]$	Intervalle entier entre n et m inclus
$\|\mathscr{X}\|$	Cardinal de $\mathscr{X}$
$\varnothing$	Ensemble vide

Vecteurs

$\overrightarrow{x}$	Vecteur
$(\ldots)$	Vecteur
x_i	$i^{\text{ème}}$ composante du vecteur $\overrightarrow{x}$
$\overrightarrow{x}^{\uparrow}$	Vecteur des composantes de $\overrightarrow{x}$ ordonnées dans l'ordre croissant
$\overrightarrow{x}^{\uparrow}_i$	$i^{\text{ème}}$ composante du vecteur $\overrightarrow{x}^{\uparrow}$
$\overrightarrow{x}^{\downarrow}$	Vecteur des composantes de $\overrightarrow{x}$ ordonnées dans l'ordre décroissant
$\overrightarrow{x}^{\downarrow}_i$	$i^{\text{ème}}$ composante du vecteur $\overrightarrow{x}^{\downarrow}$
$\overline{x}$	moyenne des composantes du vecteur $\overrightarrow{x}$

Raisonnement mathématique

$\forall$, $\exists$	Quantificateurs universel et existentiel non logiques
$\Leftarrow$, $\Rightarrow$	Implication non logique
$\Leftrightarrow$	Équivalence non logique

Relations binaires

$\Re$	Relation binaire quelconque
$x \Re y$	$\Re(x, y)$

$(\preceq,\succeq),(\leq,\geq)$	Préordres (relations binaires réflexives et transitives)
$\prec,\succ$	Préférence stricte : $x \prec y \Leftrightarrow (x \preceq y$ et $y \not\preceq x)$
$\sim$	Indifférence : $x \sim y \Leftrightarrow (x \preceq y$ et $y \preceq x)$
$\bot,\top$	Éléments minimum et maximum vis-à-vis d'un préordre
$\max_{\succeq}$	Ensemble des éléments non dominés vis-à-vis de $\succeq$
$[x]_{\sim}$	Classe d'équivalence de x vis-à-vis de la relation $\sim$
$\mathscr{E}/\sim$	Ensemble quotient de $\mathscr{E}$ pour $\sim$
Logique et contraintes	
$\mathbf{x}$	Variable binaire (booléenne) ou non binaire
$\underline{\mathbf{x}}$, $\overline{\mathbf{x}}$	valeurs minimales et maximales du domaine de $\mathbf{x}$
$\underline{\mathbf{x}} \leftarrow \alpha$, $\overline{\mathbf{x}} \leftarrow \alpha$, $\mathbf{x} \leftarrow \alpha$	modifications du domaine de $\mathbf{x}$ (dans les algorithmes)
$\neg$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$	Connecteurs logiques non, ou, et, implication, équivalence
$\mathbf{x}\ ?\ \mathbf{y_1} : \mathbf{y_0}$	opérateur logique if-then-else
$Mod(\varphi)$	Ensemble des modèles de φ
$v \models \varphi$	$v \in Mod(\varphi)$
$MaxCons(\Delta,\beta)$	ensemble de tous les sous-ensembles maximaux β-consistants de Δ
$MaxCons(\Delta)$	$MaxCons(\Delta,\top)$
$\mid\!\sim^{\forall}$	Symbole de conséquence sceptique
$Inst(\mathscr{X})$	Ensemble des instanciations (ou interprétations) sur $\mathscr{X}$
$(\mathbf{x_1} : v(\mathbf{x_1}), \ldots, \mathbf{x_n} : v(\mathbf{x_n}))$	Instanciation (notation explicite)
$\langle v_x, v_y \rangle$	Concaténation de deux instanciations
$(\mathbf{x_1} : \mathscr{D}_{\mathbf{x_1}}, \ldots, \mathbf{x_n} : \mathscr{D}_{\mathbf{x_n}})$	Fonction de domaine (notation explicite)
$\langle \mathcal{D}_1, \mathcal{D}_2 \rangle$	Unification de deux fonctions de domaine
$v_{\downarrow \mathscr{S}}$	Projection d'une instanciation sur un ensemble
Fonctions, limites	
$\lvert x \rvert$	Valeur absolue de x
$sgn(x)$	Fonction signe : $\forall x \in \mathbb{R}^{\star},\ sgn(x) = \frac{x}{\lvert x \rvert}$
$\log$	Fonction logarithme népérien
Id	Fonction identité
$f(x) \xrightarrow[x \to a]{} b$	b est la limite de $f(x)$ lorsque x tend vers a
$x \to a^+$, $x \to a^-$	x tend vers a par valeurs positives, négatives
$f(x) = o_{x \to a}(g(x))$	$\frac{f(x)}{g(x)} \xrightarrow[x \to a]{} 0$
Complexité computationnelle	
$\mathcal{P}$	Problème de décision quelconque.
co-$\mathcal{P}$	Complémentaire du problème $\mathcal{P}$.
Probabilités	
$X = (x_1, A_1; \ldots; x_n, A_n)$	Variable aléatoire
$(x_1, p_1; \ldots; x_n, p_n)$	Loterie
P	Fonction de probabilité
$E(X)$	Espérance de X
Divers	
$\binom{m+n-1}{n}$	Coefficient binômial
$\mathcal{G}$	Graphe

Introduction

La juste répartition des biens — et des travaux pour le bénéfice commun — entre les membres d'une société est la garantie du bien-être des individus, de la stabilité et de la paix sociale. Cette «juste répartition» des ressources implique l'introduction d'un ensemble de règles d'allocations et de lois acceptées par tous les membres de la collectivité, l'ensemble de ces règles correspondant à ce que les membres de la société considèrent comme *équitable*. Selon cette acception, le terme d'«équité» ne véhicule donc pas nécessairement de contenu éthique ou moral, contrairement à son emploi habituel dans le langage courant [Young, 1994]. Une règle (de partage) est considérée comme équitable si elle est acceptée et jugée appropriée par l'ensemble des membres de la collectivité, au vu des besoins, statuts et contributions de l'ensemble des individus.

Il convient de bien distinguer deux facettes de l'équité. D'un côté, l'équité au sens large concerne la répartition des biens, des droits et des obligations au sein d'une société dans son ensemble. Il s'agit du domaine de prédilection des philosophes sociaux et moraux, qui s'appliquent à définir la notion de *justice sociale*. L'autre aspect de l'équité est de nature beaucoup plus locale : elle s'applique à des situations concrètes telles que l'on peut rencontrer dans la vie de tous les jours et pour lesquelles on essaie de proposer un ensemble de règles permettant de résoudre les problèmes au cas par cas. C'est à cet aspect de l'équité que nous ferons référence dans ce travail de thèse ; nous donnerons quelques exemples de telles situations concrètes dans cette introduction.

Le problème de partage et les notions qui lui sont rattachées font en fait partie de la question beaucoup plus générale de la *prise de décision collective*. Cette question, concernant le choix, parmi un ensemble d'options possibles, d'une alternative impliquant un certain nombre d'individus, a donné naissance d'une part à la théorie du choix social, concentrée sur l'analyse et l'évaluation des méthodes dédiées à la prise de décision collective, et d'autre part à l'économie du bien-être, fondée sur la mesure de la performance et la comparaison de systèmes économiques réels ou imaginaires, ainsi que sur l'analyse et la critique de politiques économiques [Arrow *et al.*, 2002]. Si les premières réflexions sur ces sujets remontent à l'antiquité (Aristote (384–322 av. J.-C.) en Grèce et Kautilya (350–283 av. J.-C.) en Inde), les théories fondatrices de ces deux domaines sont attribuées à Marie-Jean de Condorcet (1743–1794) et Jean-Charles de Borda (1733–1799) pour la théorie du vote et du choix social, concentrée essentiellement sur les propriétés des systèmes électoraux et l'élection des comités, et à Jeremy Bentham (1748–1832) pour l'économie du bien-être, centrée sur la notion de fonction d'utilité collective. Plus récemment, les travaux fondamentaux de Kenneth J. Arrow (né en 1921) ont tenté une unification de ces deux domaines, centrée sur l'étude de l'existence *théorique* d'un processus d'agrégation raisonnable des préférences individuelles, fondé sur un ensemble d'axiomes.

Parallèlement au développement du choix social et de l'économie du bien-être, centrés sur l'interaction d'individus aboutissant à un choix collectif, s'est développée une autre discipline d'importance : la théorie de la décision individuelle. Cette discipline, située à l'interface entre l'économie, la gestion, la psychologie, la statistique et les mathématiques, est axée sur l'ensemble des processus qui permettent à un être humain de prendre une décision concernant un ensemble d'alternatives, dans un contexte pouvant être incertain ou mal connu. Ce domaine, fondé sur la notion de préférences individuelles, est lié au développement historique de la théorie des jeux (dont on attribue la fondation à John von Neumann (1903–1957) et Oskar Morgenstern (1902–1977)) et, dans une moindre mesure et de manière plus tardive, à celui de la recherche opérationnelle.

Si l'intelligence artificielle et la théorie de la décision individuelle sont historiquement liées, en revanche il n'en est pas de même pour la décision collective, qui est traditionnellement le domaine réservé des philosophes, des psychologues, des économistes théoriques ou des mathématiciens, et dont le rapprochement avec la communauté informatique est relativement récente. Cette rencontre est particulièrement intéressante. Alors que du côté de la communauté du choix social, la plupart des travaux sont concentrés sur l'axiomatisation de propriétés liées à la décision collective et sur la recherche de procédures permettant d'aboutir à une décision vérifiant un critère donné, peu de travaux se sont intéressés à la *difficulté* liée au calcul de la décision optimale, et encore moins à son *implantation*, ou en d'autres termes à sa traduction *algorithmique*. D'un autre côté, si ces notions sont à la base de la recherche en intelligence artificielle et en informatique, les travaux de ces derniers domaines sont la plupart du temps fondés sur l'optimisation d'un critère additif ou linéaire, et se préoccupent rarement de notions telles que l'équité, qui sont pourtant centrales dans le domaine du choix social. Outre les notions d'algorithmique et de complexité liées à la décision collective et individuelle, le rapprochement de ces communautés a fait émerger un champ de recherche complètement nouveau et prometteur, concernant les problèmes de décision sur des domaines finis et combinatoires. L'aspect combinatoire, pourtant crucial dans de nombreux problèmes de décision sur des domaines finis, a été jusqu'ici relativement négligé dans le domaine du choix social.

Le rapprochement récent du domaine de la décision au sens large avec l'informatique et intelligence artificielle a donc fait émerger de nouvelles problématiques communes. Ces problématiques sont regroupées sous le terme de *choix social computationnel*. Notons que ce rapprochement n'a rien de surprenant. D'un côté, le domaine de l'intelligence artificielle est par essence porté sur la compréhension, la modélisation et la formalisation des comportements humains[1], et quoi de plus humain que l'ensemble des processus aboutissant au choix d'une décision collective ; d'un autre côté, les nouveaux terrains d'investigation de la communauté du choix social ont fait émerger un certain nombre de problèmes — notamment liés à la représentation des préférences et à la prise en compte de domaines combinatoires — sur lesquels les chercheurs en informatique et intelligence artificielle peuvent apporter des outils et des réponses.

* * * * *

Le travail que nous allons présenter ici est issu de ce rapprochement entre communautés. Il sera plus précisément axé sur le problème de partage équitable. Si les fondements théoriques de l'étude des problèmes de partage s'appuient sur les bases solides du choix social et de l'économie du bien-être, le partage a des spécificités qui font de lui un domaine d'étude interdisciplinaire à part entière. Son omniprésence dans des applications concrètes, que ce soit dans le domaine industriel, institutionnel, ou domestique, explique l'abondance de littérature qui lui est consacrée, que ce soit — tout comme pour la décision collective — dans le domaine de la philosophie, dans le domaine de l'économie, ou plus récemment dans le domaine de l'informatique et de l'intelligence artificielle.

Derrière le terme de «partage» se dessine une notion aux contours flous, polymorphe, et pouvant englober tout un ensemble de problèmes, allant de l'allocation de tâches entre des machines ou des individus à la division d'un territoire entre plusieurs communautés, en passant par le partage d'objets

[1]Au moins pour les partisans de la vision «forte» de l'intelligence artificielle.

entre des agents. Il convient donc dans un premier temps de proposer une définition générique du problème de partage, afin d'en poser les limites :

Définition 1 (Problème de partage) *Un problème de partage est un problème de décision ou d'optimisation particulier défini comme suit :*

Des entrées
- ▷ *une* ressource *commune limitée,*
- ▷ *un ensemble fini $\mathscr{N}$ d'*agents *demandeurs de la ressource,*
- ▷ *un ensemble de* contraintes *sur la ressource,*
- ▷ *un* critère *d'optimisation ou de décision.*

Une sortie *Une allocation d'une partie ou de la totalité de la ressource à chaque agent qui vérifie les contraintes sur la ressource et qui optimise ou vérifie le critère donné.*

Dans cette définition, la ressource joue un rôle central. Elle peut prendre des formes très variées : ensemble fini d'objets physiques, quantité divisible finie, ensemble de quantité finies et divisibles, travaux[2]... Le point commun entre toutes les formes que peut prendre la ressource est qu'elle est toujours disponible en quantité *finie*. En outre, un certain nombre de facteurs extérieurs physiques ou légaux peuvent limiter l'ensemble des allocations possibles de la ressource : ce sont ces facteurs qui sont regroupés sous le terme de «contraintes». Les agents impliqués dans la définition du problème de partage sont «demandeurs» de la ressource dans le sens où ils expriment des préférences sur le partage de cette ressource. Le terme «agent» peut être pris au sens large : en particulier, il peut s'agir d'agents artificiels comme des machines auxquelles doivent être allouées un certain nombre de tâches. Il conviendra dans ce dernier cas d'être très prudent sur l'emploi de termes subjectifs tels que l'équité ou la justice morale afin de ne pas tomber dans le piège anthropomorphique : appliqués à des machines, ces concepts ne peuvent avoir une signification plus profonde que celles de modèles formels appliqués à des cas concrets. Enfin, le critère d'optimisation ou de décision fait bien entendu référence à ce que la collectivité juge comme étant un partage «adéquat» : autrement dit, ce critère concerne l'*équité* de la décision, mais aussi l'*efficacité*, ce dernier terme faisant référence au fait que la ressource ne doit pas être sous-exploitée.

Avant d'entrer dans le vif du sujet, de délimiter les contours de ce travail de thèse, et de détailler l'organisation du manuscrit, nous allons introduire un certain nombre d'exemples concrets de problèmes de partage réels. L'objectif de l'introduction de ces exemples est double. Tout d'abord, nous cherchons à mettre en valeur l'omniprésence de ces problèmes dans le monde réel. En outre, ces applications, dont la première qui sera présentée est l'inspiratrice de ce travail de thèse, nous serviront à illustrer les différents chapitres de ce manuscrit, et seront, pour les applications 1, 4 et 5 à la base des expérimentations présentées dans le chapitre 6.

Application 1 (Partage de ressources satellitaires) En raison de leur coût prohibitif, les projets spatiaux de grande envergure sont souvent cofinancés, puis coexploités par plusieurs agents (pays, entreprises, agences civiles ou militaires ...). C'est le cas en particulier des constellations de satellites d'observation de la Terre telles que la constellation Pléiades, projet orchestré par le *Centre National d'Études Spatiales.*

La mission de ce type de satellites consiste, comme l'illustre la figure 1, à acquérir des photographies de la Terre, en réponse à des demandes de prises de vues déposées par les agent auprès d'un centre de planification commun situé au sol. Ces demandes sont d'importances inégales pour les agents, ce qui s'exprime par l'association d'un poids numérique à chaque requête. En outre, les demandes peuvent concerner des prises de vue simples, ou des prises de vue complexes demandant plusieurs acquisitions successives, comme par exemple les prises de vue stéréographiques nécessitant

[2]Notons que le problème de partage de travaux ou de corvées ne se distingue pas fondamentalement du problème de partage de ressources : il suffit de considérer le travail comme une ressource «négative».

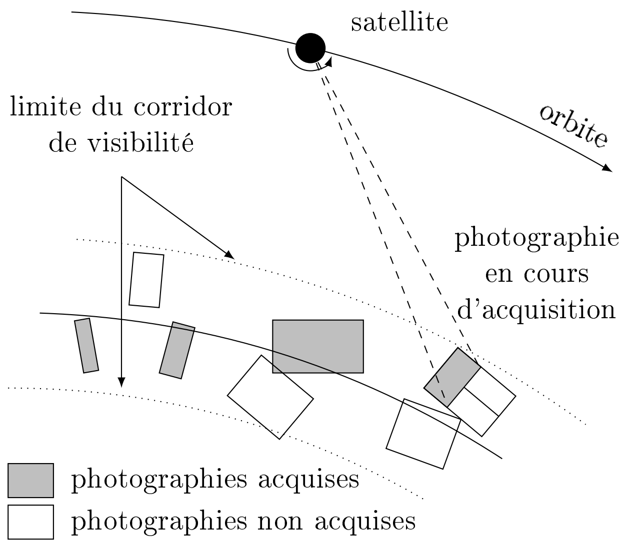


Figure 1 — Acquisition d'une photographie par un satellite d'observation de la Terre.

deux acquisitions, ou encore les prises de vues concernant des zones géographiques plus larges que la fauchée de l'instrument. Notons encore qu'une prise de vue peut être réalisée de plusieurs manières différentes, correspondant par exemple à plusieurs révolutions successives d'un satellite, ou encore à l'acquisition par deux satellites différents de la constellation.

Les demandes sont valables pour un jour donné, et la planification globale des prises de vue de tous les satellites de la constellation est organisée par intervalles de temps successifs. Le centre de planification détermine donc, parmi les demandes concernant un jour donné, l'ensemble des demandes qui seront satisfaites, c'est-à-dire l'ensemble des photographies qui seront acquises ce jour-là par la constellation. Cet ensemble de demandes satisfaites constitue une *allocation* journalière des demandes aux agents.

Les contraintes physiques d'exploitation et le nombre important de demandes concernant certaines zones engendrent des conflits entre demandes. Il est donc en général impossible de satisfaire simultanément toutes les demandes déposées pour un jour donné. Autrement dit, seul un sous-ensemble des demandes pourront être satisfaites. Toutes ces contraintes définissent l'ensemble des *allocations admissibles*. Voici quelques ordres de grandeur concernant le problème réel. Les agents sont entre 3 et 6. Plusieurs centaines de demandes sont candidates chaque jour, parmi lesquelles 100 à 200 seront satisfaites.

Tous les agents n'ont pas contribué de manière égale au financement de la constellation, ce qui se traduit donc par un «droit de retour sur investissement» différent pour chaque agent. Prenant en compte ces contraintes, l'exploitation de la constellation de satellites doit se faire de manière :

- ▷ *équitable*, dans la mesure où chaque agent attend un retour sur investissement en rapport avec sa contribution financière dans le projet ;
- ▷ *efficace*, dans le sens où la constellation ne doit pas être sous-exploitée.

On pourra trouver une description complète et détaillée du problème Pléiades dans [Lemaître *et al.*, 2002], et une description simplifiée du problème dans [Lemaître *et al.*, 1999; Fargier *et al.*, 2004a]. Les aspects algorithmiques dédiés à ce problème ont de plus été étudiés dans [Bianchessi *et al.*, 2007].

Application 2 (Partage de réseaux informatiques) Un domaine très concerné par la problématique du partage est celui des réseaux de télécommunication, et plus particulièrement celui des réseaux informatiques. Ce domaine concerne en effet par essence l'allocation d'une ressource critique, la bande-passante, à un ensemble d'utilisateurs demandeurs de la ressource, par le biais d'applications logicielles utilisatrices du réseau. Dans ce type d'applications, la notion d'équité est à la fois fondamentale et mal définie, car on veut garantir une qualité de service équivalente à tous les utilisateurs, alors que dans le même temps, les applications utilisatrices du réseau sont très hétérogènes (applications élastiques comme les clients e-mail ou FTP, applications temps-réel comme la lecture de flux vidéo en ligne, etc.).

De nombreux travaux portent sur la notion d'équité dans le partage de réseaux informatiques. Cette notion peut intervenir à différents niveaux : dans des niveaux très proches de la couche matérielle (systèmes de queues équitables pour garantir le traitement égal de chaque datagramme), ou dans les couches logicielles (l'égalité du temps d'accès au réseau peut être critique dans certaines applications comme les enchères en ligne).

On pourra trouver dans [Denda *et al.*, 2000] une étude générale du problème de l'équité dans les réseaux informatiques. Le problème de l'optimisation automatique du routage dans les réseaux informatiques a été étudiée dans [Frei *et al.*, 2005], qui aborde le problème du point de vue de la planification estimée du trafic, et non sur demande comme la plupart des approches.

Application 3 (Partage de l'espace aérien) L'augmentation démesurée du trafic aérien depuis une vingtaine d'années, et la poursuite prévue de son augmentation pour les prochaines décennies pose de manière de plus en plus critique des problèmes de congestion au niveau des aéroports et dans l'espace aérien. Actuellement, le facteur limitant principal de la capacité de l'espace aérien est donné par la charge de travail des contrôleurs, qui sont chargés de réguler l'écoulement du trafic dans des conditions optimales de sécurité.

La congestion du secteur aérien peut être diminuée de deux manières : soit par augmentation des infrastructures aéroportuaires, une meilleure division de l'espace aérien en secteurs, et une amélioration technologique des systèmes de surveillance et de navigation, ce qui revient à augmenter la ressource disponible en l'utilisant mieux, soit par une régulation de la demande par exemple par des procédures d'allocation de créneaux de décollage et d'atterrissage dans les aéroports.

Le problème de diminution de la congestion de l'espace aérien est un exemple typique de problème d'allocation de ressources, dans lequel l'équité joue un rôle crucial. Il est en effet nécessaire de répartir les problèmes liés à la répartition des créneaux (tels que les retards dans les vols) de manière équitable entre les compagnies aériennes. Ce problème, comme la plupart des problèmes de régulation de trafic, est traité par des mécanismes incitatifs tels que des systèmes de taxes variables.

On pourra se référer, pour plus de détails sur ce problème particulier de partage de l'espace aérien et des ressources aéroportuaires à la thèse [Deschinkel, 2001], ainsi qu'aux articles [Jonker *et al.*, 2005; Faltings, 2005].

Application 4 (Allocation de sujets à des étudiants) Dans de nombreuses institutions telles que des écoles ou des universités se posent le problème d'affectation de sujets, de travaux ou de modules d'enseignement à des étudiants. L'exemple concret que nous avons à disposition est celui de l'affectation de sujets de travaux expérimentaux (abrégé en «TREX») à des étudiants de SUPAÉRO : un ensemble de sujets doit être réparti entre des binômes d'étudiants ayant formulé des préférences sur ces sujets, sous la forme d'un ordre total. Chaque binôme doit être pourvu de deux sujets, et la répartition de ces sujets doit être équitable (en dehors de toute considération de mérite ou de niveau scolaire). Le problème particulier est décrit dans la section 6.4 du chapitre 6.

Ce problème est particulièrement intéressant, car on le retrouve sous des formes diverses dans de nombreuses situations du monde réel, impliquant une allocation d'objets à des agents dans laquelle chaque agent doit avoir un nombre fixe d'objets (ce sont des problèmes du type «couplage agents-objets»). Ces problèmes sont typiques du domaine de l'affectation de tâches ou des problèmes d'emplois du temps équitables.

Application 5 (Enchères combinatoires) L'avènement de l'ère Internet a entraîné le développement d'un nombre important de systèmes de négociation en ligne et de commerce électronique. Ces systèmes sont en général fondés sur des implantations logicielles de divers mécanismes d'enchères. Le fonctionnement de ces mécanismes repose sur deux phases distinctes : lors de la phase de *mise*, une autorité centrale (le commissaire-priseur) recueille l'ensemble des mises des agents enchérisseurs, et lors de la phase de *détermination du gagnant* (*winner determination*), le commissaire-priseur calcule l'allocation optimale des objets aux agents, correspondant à la maximisation de son revenu. Parmi ces mécanismes d'enchères, ceux pour lesquels les agents peuvent miser sur des lots d'objets (et non sur des objets simples) sont d'un intérêt particulier, car le *Winner Determination Problem* y est très complexe. Ce type d'enchères sont appelées les *enchères combinatoires*.

Le domaine des enchères combinatoires est le sujet d'un nombre croissant de publications dans le domaine de l'intelligence artificielle [Cramton *et al.*, 2006; Sandholm, 1999, 2002; Rothkopf *et al.*, 1998; Lehmann *et al.*, 1999]. Cet engouement est dû d'une part au fait que le *Winner Determination Problem* est un problème concret ayant de nombreuses applications, et d'autre part à sa complexité algorithmique (malgré sa relative simplicité apparente), et la complexité liée à l'expression des mises par les agents.

Notons que la notion d'équité n'est pas au centre de ce problème, puisque l'on ne cherche pas à contenter les agents sur des règles communes acceptées par tous, mais à maximiser le revenu du commissaire-priseur. Toutefois l'équité n'est pas complètement absente du problème, et peut se présenter sous des formes diverses : équité de la procédure d'expression des mises, traitement égal des agents garanti par l'anonymat des lots, etc.

* * * * *

Comme nous l'avons fait remarquer, le rapprochement de la communauté de la décision avec celle de l'informatique et de l'intelligence artificielle a fait émerger un certain nombre de problèmes nouveaux, en particulier dans le domaine du partage équitable, nécessitant une approche pluridisciplinaire : représentation compacte et élicitation des préférences sur des domaines combinatoires, complexité, implantation des modèles et algorithmique liée aux problèmes de partage. L'objectif de cette thèse est d'étudier ces problématiques, dans le cadre particulier du partage de biens indivisibles.

Ce travail est centré sur trois aspects principaux :

- ▷ la *modélisation* et l'introduction d'un cadre formel permettant d'englober les différents aspects du problème de partage ;
- ▷ l'étude de la représentation compacte du problème de partage sous différentes formes et l'analyse de la *complexité* théorique liée à cette représentation compacte ;
- ▷ l'étude *algorithmique* du problème de partage équitable de biens indivisibles, sous le point de vue d'un critère particulier, l'ordre de bien-être social égalitariste leximin.

Le manuscrit est divisé en trois parties, dédiées respectivement à la modélisation, à la représentation compacte et à la complexité, et enfin à l'algorithmique. L'objectif de la première partie est de poser les bases de la modélisation du problème de partage. Ces bases sont introduites dans le premier chapitre, construit de manière à mettre en avant l'aspect polymorphe et multifacette du problème de partage. Ce chapitre est fondé sur l'ensemble de la littérature concernant la théorie de la décision et la théorie du choix social et du *welfarisme* cardinal. Le second chapitre de cette première partie a un statut un peu à part dans l'ensemble de la thèse, puisqu'il est centré sur le problème particulier de l'introduction de droits exogènes dans le modèle du *welfarisme* cardinal introduit au premier chapitre, et constitue la première contribution de ce travail de thèse. La deuxième partie est centrée sur l'aspect combinatoire des problèmes de partage. Elle se divise donc logiquement en deux chapitres, traitant respectivement de la problématique de la représentation compacte, indissociable de la notion de problème de décision ou d'optimisation sur un espace combinatoire, et de la complexité, liée bien entendu à la représentation compacte introduite. Le chapitre sur la complexité, lié à l'introduction de deux langages de représentation compacte du partage, constitue la deuxième contribution de ce travail de thèse. Enfin, la troisième et dernière partie, qui constitue la troisième contribution de la thèse, est centrée sur l'étude d'un problème algorithmique particulier lié au partage équitable : celui du calcul de solutions leximin-optimales. Nous introduisons dans le chapitre 5 un ensemble d'algorithmes dédiés à ce problème. La comparaison expérimentale de ces algorithmes sur des problèmes réalistes est détaillée dans le chapitre 6.

On pourra éventuellement avoir une lecture non linéaire du manuscrit, cependant la bonne compréhension de certains chapitres nécessite un certain nombre de notions et de notations introduites dans les chapitres précédents. Nous avons présenté sur la figure 2 les relations de dépendance entre les chapitres du manuscrit, à considérer comme un guide de lecture. Dans cette figure, les arcs en traits épais figurent une dépendance forte entre les chapitres (la compréhension du second est sérieusement compromise sans la lecture du premier) ; les arcs en traits fins figurent une dépendance faible entre les chapitres (la compréhension du second dépend de la connaissance de quelques notions de base introduites dans le premier) ; enfin, les arcs en pointillés figurent des références ponctuelles entre les chapitres.

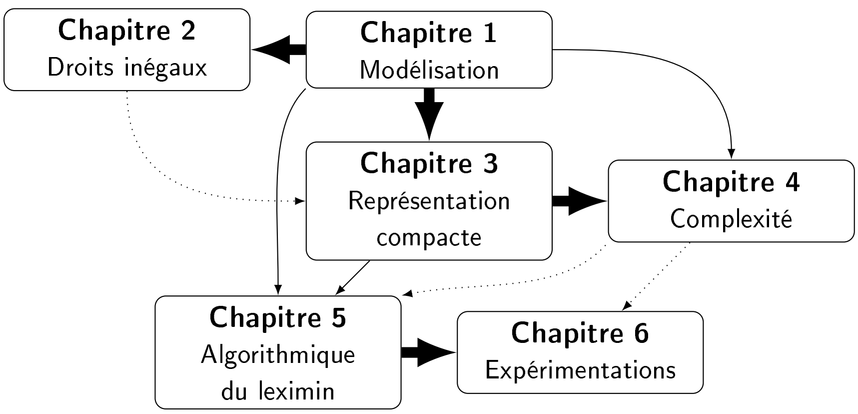
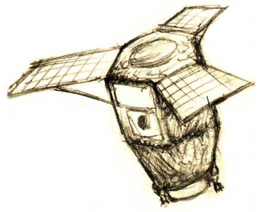


Figure 2 — Graphe de dépendances entre chapitres.

Première partie

Modélisation



Chapitre 1
Partage et décision collective

La définition générique du problème de partage présentée dans le chapitre d'introduction englobe, comme nous l'avons fait remarquer, un ensemble hétérogène de problèmes. Nous allons introduire dans ce chapitre les principales notions liées à la définition des problèmes de partage. L'objectif est de présenter un aperçu étendu de ces différents problèmes. L'introduction de ces notions nous permettra ainsi de limiter l'étendue de notre étude à un cadre bien précis, et nous fournira les bases nécessaires à la modélisation de ces problèmes.

La vue d'ensemble du problème de partage que nous proposons ici est centrée sur les questions qui suivent. Sur quel type de ressource travaille-t-on exactement ? Comment cette ressource est-elle allouée ? Comment les agents expriment-ils leurs préférences ? Sur quel(s) critère(s) peut-on juger de la qualité d'un partage ? La manière d'aborder ces questions est inspirée de l'article [Chevaleyre *et al.*, 2006a] qui présente une vue d'ensemble générale sur les problèmes d'allocation de ressources multiagent.

1.1 La ressource

La ressource, qui est par essence en quantité limitée, est le point central des problèmes de partage. Sa nature et les contraintes qui en restreignent l'allocation déterminent le type du problème à traiter. Nous supposons avant toute chose que nous sommes dans un contexte *statique*, c'est-à-dire que la ressource n'évolue pas dans le temps (ce qui n'est pas le cas par exemple pour des denrées périssables, du carburant, ou encore les prises de vues du satellite Pléiades, qui perdent leur valeur si elles sont délivrées trop tard). Cette dernière approximation est cependant raisonnable pour la plupart des problèmes de partage réels.

1.1.1 La nature de la ressource

La première distinction entre les différents problèmes de partage porte généralement sur le type de la ressource elle-même. On distingue traditionnellement deux grands types de ressource : ressource *continue*, ou ressource *discrète*. Passant outre la farouche bataille épistémologique sur l'existence du continu dans le monde réel, nous nous contenterons d'une définition simple et intuitive de cette dualité. Alors qu'une ressource continue peut être *a priori* divisée indéfiniment (du moins à notre niveau de modélisation), à l'instar d'une quantité numérique comme un volume, une masse, ou une quantité d'argent, une ressource discrète ne peut être divisée qu'un nombre fini de fois, jusqu'à arriver à un ensemble d'atomes indivisibles.

On peut faire, parmi les problèmes faisant intervenir une ressource discrète, un autre niveau de distinction, selon que la ressource est divisible (de manière finie, donc) ou non. En toute rigueur, ces deux cas ne sont pas différents l'un de l'autre, puisque dans le cas d'une ressource divisible, on peut travailler sur l'ensemble des atomes comme si l'on travaillait sur une ressource indivisible. Il s'agit donc plus d'une différence d'approche du problème que d'une propriété intrinsèque de la ressource. Cette remarque s'applique de même à la distinction entre les problèmes dits à unités multiples, qui font intervenir plusieurs instances de chaque objet, et les problèmes à unité simple, ne faisant intervenir qu'une seule instance de chaque objet. Dans la suite du document, on ne distinguera plus ressource discrète et ensemble de ressources indivisibles.

Certains problèmes peuvent faire intervenir une ressource mixte (des objets et un volume de liquide par exemple). Parmi ces problèmes figurent notamment ceux qui concernent l'allocation de ressources indivisibles avec compensation monétaire entre les agents. Dans ce genre de problèmes, la monnaie intervient comme une ressource continue ayant un statut spécial : elle ne fait pas partie de la ressource à partager à proprement parler, mais peut être utilisé sous forme de transferts entre agents pour compenser l'inéquité du partage. Une solution du problème de partage est donc dans ce cas-là une allocation de la ressource aux agents, et un ensemble de transferts monétaires entre les agents.

À la lueur de ces considérations, nous pouvons donc proposer une définition formelle de la notion de ressource, fondée sur la dichotomie ressource continue / ressource discrète :

Définition 1.1 (Ressource)
- ▷ *Une ressource* continue *est un ensemble en bijection avec* $[0, 1]$.
- ▷ *Une ressource* discrète *est un ensemble fini de ressources* indivisibles $\{o_1, \ldots, o_p\}$.

Nous pouvons à présent définir de manière formelle la notion de partage, ou d'allocation[1] :

Définition 1.2 (Partage) *Soient $\mathscr{R}$ une ressource et $\mathscr{N}$ un ensemble fini de n agents. Un* partage *de $\mathscr{R}$ entre les agents de $\mathscr{N}$ est un tuple $\overrightarrow{\pi} \in (\wp(\mathscr{R}))^n$. La composante π_i est appelée la* part *de l'agent i.*

En d'autres termes, un partage est simplement défini comme un vecteur de parts, une part étant une partie de la ressource revenant à un agent particulier. Notons que nous ne supposons pas ici que les parts sont disjointes deux à deux. Ce sera le cas seulement si la contrainte de préemption est présente, comme nous allons le voir un peu plus loin.

La nature de la ressource est une donnée cruciale dans la définition du problème de partage à traiter. Selon la nature de la ressource, on aboutit à des types de problèmes très différents dans leur difficulté, dans leur modélisation et dans leur traitement. À la lueur de la littérature sur le sujet, nous pouvons mettre en évidence les classes de problèmes suivantes, caractérisées par les types de ressources que les problèmes font intervenir.

- ▷ Une ressource continue et homogène, mais les agents ont un pouvoir imparfait. Précisons ce que l'on entend par là. Puisque la ressource est continue et homogène, les préférences des agents s'expriment comme une fonction de la quantité de ressource qu'ils reçoivent : trouver un partage équitable (si tant est que les agents ont un droit d'accès identique sur la ressource) revient dans ce cas-là à un problème d'optimisation continue sous contrainte : on cherche à égaliser un certain nombre de fonctions sous la contrainte que la somme des quantités allouées aux agents soit égale à la quantité de ressource disponible. Dans la plupart des problèmes, on suppose en plus que les préférences des agents sont identiques et dépendent linéairement de la quantité de ressource reçue, rendant le problème trivial (il suffit d'allouer à chaque agent

[1] Par la suite, nous emploierons les deux termes de manière interchangeable

le $n^{\text{ème}}$ de la ressource). La difficulté vient alors du fait que l'on est incapable de découper la ressource de manière très précise, et que les agents ont des perceptions différentes sur la quantité de ressource contenue dans une part (ainsi, un agent pourra estimer que sa part représente moins du $n^{\text{ème}}$ de la ressource, se sentant donc lésé dans le partage, alors qu'un autre agent estimera que cette même part représente une fraction supérieure au $n^{\text{ème}}$ de la ressource). Le nœud du problème se résume donc à la recherche d'une procédure permettant aux agents d'aboutir de manière décentralisée à un partage de la ressource que chacun estime juste. L'exemple typique d'application à ce problème est celui du partage d'un gâteau homogène (un exemple jouet dont l'intérêt dépasse largement l'aspect simpliste apparent du problème et qui mobilise de nombreux économistes et mathématiciens). Un autre exemple d'application concerne le partage de territoires ou de droits d'exploitation sur des ressources naturelles, si l'on considère en première approximation que ces ressources sont homogènes. Ce type de problème a abouti à la mise au point de méthodes du type «je coupe, tu choisis» (*divide-and-choose*), généralisées à n agents. On pourra lire avec profit [Brams et Taylor, 1996, 2000; Robertson et Webb, 1998] sur l'étude des procédures d'allocation de ressource divisible et homogène.

▷ La ressource est continue mais hétérogène, les préférences des agents, qui ont toujours un pouvoir imparfait, sont en rapport avec cette hétérogénéité. Ici la difficulté vient toujours du fait que les agents sont incapables de partager la ressource de manière parfaite, ajoutant à cela la difficulté supplémentaire liée à l'hétérogénéité de la ressource. Il existe de multiples exemples de problèmes de partage de ressources continues hétérogènes : partage de territoires, partage de n ressources continues, ... Ces problèmes sont étudiés en détail dans [Brams et Taylor, 1996, 2000], donnant lieu par exemple à des procédures telles que *Adjusted Winner*, utilisées dans des problèmes concrets. Notons que le problème de partage de gâteau est encore une fois considéré comme une métaphore de base pour désigner une ressource continue et hétérogène quelconque : ce genre de problèmes a l'avantage d'être simple à comprendre et illustratif. On pourra consulter l'article [Brams *et al.*, 2006] pour avoir un aperçu des procédures utilisées dans ce cas pour garantir l'équité ou encore l'absence d'envie.

▷ La ressource est discrète, mais des compensations monétaires sont possibles. Il s'agit de l'un des problèmes les plus étudiés dans la littérature sur le problème de partage (encore une fois, on pourra consulter les ouvrages de référence cités ci-dessus), car il englobe un grand nombre de problèmes réels pouvant s'avérer délicats, dans lesquels le besoin d'équité est crucial : problèmes de partage d'objets après divorce, héritage, ... Deux aspects sont concernés par ces problèmes : partage des objets indivisibles eux-mêmes, et calcul des compensations financières nécessaires au rétablissement de l'équité. L'objectif est donc de mettre au point des procédures permettant d'une part de se rapprocher le plus possible de l'équité lors du partage d'objet (équité dans le sens où chacun estime avoir sa «juste part»), et d'autre part d'atteindre effectivement l'équité à l'aide de compensations monétaires. Étant donné le contexte dans lequel ces procédures sont en général appliquées, elle nécessitent de plus d'être résistantes aux manipulations des agents, dans le sens où elle doivent dissuader les individus de falsifier leurs préférences. Un exemple de procédure classique étudiée dans ce contexte particulier est celle de Knaster.

▷ La ressource est discrète, mais les compensations monétaires sont impossibles. Il est rare dans ce type de problème que l'on puisse atteindre un état d'équité parfaite. On va donc chercher à s'en approcher le plus possible. On peut avoir affaire dans ce cas à deux types de problème très différents :

- Si le nombre d'objets en jeu est faible (dans le cas extrême, il peut n'y avoir qu'un seul objet, par exemple un rein à attribuer à un patient en attente de greffe), et que les agents ont des préférences simples, il s'agit d'un problème éthique ou moral, qui est de savoir à qui attribuer la ressource [Young, 1994]. Entrent en jeu des considérations telles que le besoin, le mérite, l'adéquation ou la priorité. Nous aurons l'occasion d'y revenir un peu plus loin.

- Si le nombre d'objets en jeu est élevé et que les agents ont des préférences complexes sur les objets, s'exprimant par des dépendances, on se trouve typiquement dans un cas de problème d'allocation combinatoire. Ici, la difficulté du problème est liée à l'explosion combinatoire due à la structure de l'espace des partages admissibles, comme nous le verrons au chapitre 3. C'est le domaine privilégié des problèmes d'enchères combinatoires[2] [Cramton *et al.*, 2006; Sandholm, 1999, 2002; Rothkopf *et al.*, 1998; Lehmann *et al.*, 1999].

Comme nous pouvons donc le constater, l'étude des problèmes de partage de ressource continue ou de ressource discrète avec compensations monétaires est une discipline traditionnelle et largement étudiée dans le domaine économique. En revanche, les quelques travaux récents en informatique et intelligence artificielle concernant le domaine du partage sont plutôt centrés sur des ressources discrètes.

Concluons notre discussion sur la nature de la ressource en remarquant que la limite entre les deux types de ressource est parfois difficile à appréhender, ou peut dépendre de l'approche utilisée pour modéliser ou résoudre le problème. Ainsi par exemple un problème impliquant une ressource continue peut être traité par discrétisation, c'est-à-dire en divisant la ressource en un ensemble de parts indivisibles, le processus d'allocation portant ensuite sur cet ensemble de parts : dans un problème de partage d'une réserve d'eau d'une capacité de 10000 litres, les agents peuvent s'accorder sur le partage de 200 unités de 50 litres plutôt que sur le partage au millilitre près de la ressource. Ainsi, une ressource continue peut être traitée en première approximation avec des techniques s'appliquant aux ressources discrètes, bien que ce ne soit pas toujours la méthode la plus efficace.

Dans d'autres problèmes, la ressource à partager elle-même peut dépendre de la modélisation adoptée. Considérons l'application 1 de partage d'une constellation de satellites. Cet exemple peut être abordé sous deux angles différents. Soit on considère que l'on partage l'ensemble de requêtes émises par les agents, et nous avons donc affaire à une ressource indivisible, soit on aborde le problème sous le point de vue du partage du temps d'utilisation de la constellation de satellites, le temps étant par essence une ressource continue.

1.1.2 Les contraintes d'admissibilité

Dans tout problème de partage, l'attribution de la ressource aux agents est soumise à un certain nombre de contraintes. Ces contraintes peuvent être notamment de nature physique (impossibilité par exemple d'attribuer le même objet à deux agents différents), ou encore de nature légale (impossibilité pour un agent d'acquérir plus d'une certaine quantité de ressource). Par «contrainte» nous désignerons toute condition imposée sur la ressource, qui restreint l'ensemble des allocations possibles aux agents, soit, de manière formelle :

Définition 1.3 (Contrainte d'admissibilité) *Soient $\mathscr{R}$ une ressource et $\mathscr{N}$ un ensemble fini d'agents. Une* contrainte d'admissibilité *sur l'attribution de la ressource aux agents de $\mathscr{N}$ est un sous-ensemble $C \subsetneq \wp(\mathscr{R})^n$.*

Cette définition ne fait aucune supposition sur la manière dont est exprimée la contrainte. Nous aurons l'occasion de revenir sur les langages d'expression de contraintes dans le chapitre 3. L'introduction de contraintes dans le partage nous permet de définir la notion d'*allocation admissible* :

[2] Même si, dans une certaine mesure, les compensations monétaires interviennent dans ce domaine, car un agent qui paie son lot est «compensé négativement», l'argent n'intervient pas en tant que ressource de base à partager, mais en tant que mesure d'utilité uniquement.

Définition 1.4 (Partage admissible) *Soient $\mathscr{R}$ une ressource, $\mathscr{N}$ un ensemble fini d'agents, et $\mathscr{C}$ un ensemble de contraintes. Un* partage admissible *de $\mathscr{R}$ entre les agents de $\mathscr{N}$ vis-à-vis de l'ensemble de contraintes $\mathscr{C}$ est un tuple $\overrightarrow{\pi} \in \bigcap_{C \in \mathscr{C}} C$. L'ensemble des partages admissibles sera noté $\mathscr{A}$.*

Parmi l'ensemble des contraintes possibles, une contrainte joue un rôle particulier, car elle est présente de manière naturelle dans la plupart des problèmes de partage. Il s'agit de la contrainte de préemption. Cette contrainte interdit l'allocation d'une même partie de la ressource à plusieurs agents ; en d'autres termes, elle impose aux parts des agents d'être disjointes deux à deux. Cette contrainte est présente naturellement dans la plupart des problèmes de partage, dans lesquels la ressource à allouer correspond à un bien physique et à une réalité matérielle. En revanche, certains problèmes faisant intervenir une ressource virtuelle sont affranchis de cette contrainte. Nous pouvons citer le problème Pléiades présentée dans l'application 1 : la ressource à partager est l'ensemble des prises de vue (si l'on adopte ce point de vue, et non le point de vue du partage du temps d'exploitation de la constellation), une même prise de vue pouvant être attribuée à plusieurs agents différents. Nous pouvons de même citer l'exemple d'une entreprise qui répartit les licences d'utilisation de ses logiciels entre ses employés : un même logiciel peut être utilisé par plusieurs employés différents, dans la limite du nombre de licences disponibles.

La contrainte de préemption s'exprime simplement comme suit, que la ressource soit continue ou non :

$$C_{preempt} = \{(\pi_1, \ldots, \pi_n) \mid \forall i \neq j \; \pi_i \cap \pi_j = \varnothing\}.$$

Un certain nombre d'autres contraintes apparaissent de manière naturelle dans les problèmes de partage.

▷ Les contraintes d'exclusion empêchent une certaine partie de la ressource d'être attribuée entièrement au même agent : $\mathscr{S}_{excl}$ est le sous-ensemble de la ressource $\mathscr{R}$ (ce qui correspond dans le cas discret à un sous-ensemble d'objets $\mathscr{O}_{excl}$) qui ne peut être attribué au même agent, alors $C_{excl} = \{(\pi_1, \ldots, \pi_n) \mid \forall i \; \pi_i \not\supseteq \mathscr{S}_{excl}\}$.

▷ On peut de même définir une contrainte d'exclusion globale, qui interdit l'attribution simultanée d'une partie de la ressource, même si cette partie n'est pas attribuée à un unique agent : $C_{glob\_excl} = \{(\pi_1, \ldots, \pi_n) \mid \bigcup_i \pi_i \not\supseteq \mathscr{S}_{excl}\}$.

▷ Les contraintes de volume interdisent l'attribution de plus d'un certain volume de ressource à un agent, si l'on considère qu'à la ressource est attachée une fonction volume :
  - ressource continue : soient $\mu : \mathscr{R} \to \mathbb{R}^+$ une mesure sur $\mathscr{R}$, représentant la mesure de volume, et $V_{max} \in \mathbb{R}^+$ le volume maximum autorisé par agent, $C_{vol} = \{(\pi_1, \ldots, \pi_n) \mid \forall i \mu(\pi_i) \leq V_{max}\}$;
  - ressource indivisible : soit $\nu : \mathscr{O} \to \mathbb{R}^+$ une fonction de volume, et V_{max} le volume maximum autorisé par agent, $C_{vol} = \{(\pi_1, \ldots, \pi_n) \mid \forall i \; \sum_{o \in \pi_i} \nu(o) \leq V_{max}\}$.

  Bien entendu, dans le cas indivisible, une contrainte de volume peut être traduite en un ensemble de contraintes d'exclusions (les objets dont le volume global est supérieur au volume maximal sont mutuellement exclusifs). À l'instar de la contrainte d'exclusion, on peut définir une contrainte de volume globale qui interdit l'attribution globale de plus d'un certain volume de ressource.

▷ Plus généralement, on peut définir des contraintes de dépendances entre objets, imposant par exemple l'attribution simultanée de deux objets différents à un même agent.

1.2 La notion de préférences

Le problème de partage de ressource fait intervenir une collectivité d'agents confrontés au problème épineux de la répartition de la ressource au sein de la collectivité. De ce point de vue, ce problème peut être vu comme un problème de décision (collective en l'occurrence), face à l'ensemble des alternatives (partages) possibles. Dès lors que l'on s'intéresse à la notion de décision impliquant des agents humains apparaît de manière naturelle la notion de préférences sur l'espace des alternatives. Comme le rappellent Denis Bouyssou et Philippe Vincke dans [Bouyssou et Vincke, 2006], la problématique de la modélisation des préférences intervient dans un ensemble de domaines très différents : économie, psychologie, sciences politiques, recherche opérationnelle, intelligence artificielle, ou de manière plus générale l'ensemble des domaines scientifiques dont s'inspire la théorie de la décision.

La notion de préférence intervient à deux niveaux dans les problèmes de partage, ou plus généralement dans les problèmes de décision collective.

- ▷ Au niveau individuel, chaque agent a des préférences sur l'issue du partage, représentant son propre point de vue personnel. La construction des préférences individuelles se fait généralement par un processus d'*élicitation*, et relève du domaine de la représentation et de l'expression des préférences, dont nous reparlerons dans le chapitre 3 consacré à la représentation compacte.
- ▷ Au niveau collectif, l'agrégation des préférences individuelles conduit à des préférences collectives sur l'issue du partage, reflétant l'ensemble des préférences (souvent contradictoires) individuelles. La construction des préférences communes relève du domaine de la décision collective : vote et choix social, ou théorie de l'utilitarisme, dont nous reparlerons plus loin dans ce chapitre.

1.2.1 Modélisation des préférences

Cette sous-section sera consacrée aux bases de la modélisation des préférences. Nous allons commencer par présenter l'approche classique en théorie de la décision, qui consiste à définir une structure de préférences ordinale. L'introduction et la présentation des outils de modélisation des préférences en théorie de la décision s'inspire de l'approche présentée dans des ouvrages comme [Vincke, 1989] ou dans [Bouyssou et Vincke, 2006]. Nous introduirons ensuite quelques structures de préférences classiques étendant la structure ordinale en y ajoutant des informations sur l'intensité des préférences par exemple.

1.2.1.1 Relations binaires

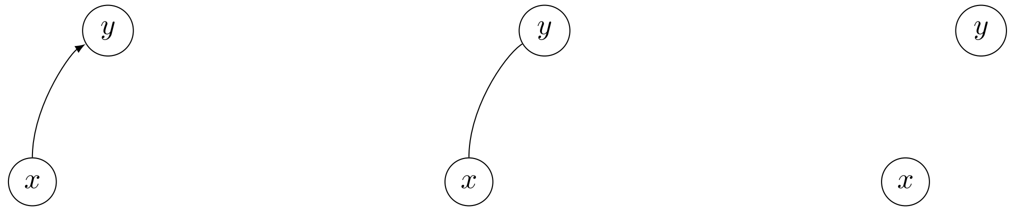
L'outil central de modélisation des préférences est la *relation binaire* :

Définition 1.5 (Relation binaire) *Étant donné un ensemble $\mathscr{E}$, une* relation binaire $\Re$ *sur $\mathscr{E}$ est un sous-ensemble du produit cartésien $\mathscr{E} \times \mathscr{E}$. On emploie habituellement la notation $x \Re y$ pour désigner $(x, y) \in \Re$.*

Une relation $\Re$ sera dite :

- ▷ *réflexive si et seulement si $x \Re x$ pour tout $x \in \mathscr{E}$;*
- ▷ *symétrique si et seulement si $x \Re y \Rightarrow y \Re x$ pour tout $(x, y) \in \mathscr{E}^2$;*
- ▷ *antisymétrique si et seulement si $x \Re y$ et $y \Re x \Rightarrow x = y$ pour tout $(x, y) \in \mathscr{E}^2$;*
- ▷ *transitive si et seulement si $x \Re y$ et $y \Re z \Rightarrow x \Re z$ pour tout $(x, y, z) \in \mathscr{E}^3$;*
- ▷ *complète si et seulement si $x \Re y$ ou $y \Re x$ pour tout $(x, y) \in \mathscr{E}^2$.*

Une relation binaire peut être représentée sous la forme d'un graphe orienté dont les sommets sont les éléments de l'ensemble $\mathscr{E}$, et les arcs sont les couples (x, y) tels que $x\Re y$ (voir figure 1.1). Afin d'alléger la notation, on représentera sous la forme d'un arc non orienté la relation entre deux éléments x et y tels que $x\Re y$ et $y\Re x$. De même, dans le cas particulier de relations réflexives (comme ce sera le cas lors de la modélisation des préférences), nous ne représenterons pas les boucles (x, x).



$x\Re y$ et non $y\Re x$. $x\Re y$ et $y\Re x$. non $x\Re y$ et non $y\Re x$.

Figure 1.1 — Représentation d'une relation binaire sous forme d'un graphe.

Parmi les relations binaires, certaines sont d'importance dans la modélisation des préférences : les relations d'équivalence, les préordres et les ordres.

Définition 1.6 (Relation d'équivalence, classe d'équivalence) *Une* relation d'équivalence $\sim$ *est une relation binaire réflexive, symétrique et transitive.*

Si $\sim$ est une relation d'équivalence sur $\mathscr{E}$, on note, pour tout élément $x \in \mathscr{E}$, $[x]_\sim$ la classe d'équivalence de x pour $\sim$, c'est-à-dire l'ensemble $\{y \in \mathscr{E} \mid x \sim y\}$. L'ensemble des classes d'équivalence de $\mathscr{E}$ pour $\sim$ est appelé ensemble quotient de $\mathscr{E}$ par $\sim$*, forme une partition de $\mathscr{E}$, et est noté $\mathscr{E}/\sim$.*

Le graphe représentant une relation d'équivalence est un graphe non orienté (à cause de la propriété de symétrie), formé d'une forêt de cliques (à cause de la propriété de transitivité). Chaque clique représente une classe d'équivalence de la relation.

Définition 1.7 (Relation d'ordre) *Une* relation d'ordre $\geq$ *est une relation binaire réflexive, transitive et antisymétrique. Si en plus la relation est complète, elle sera appelée* relation d'ordre total.

Une relation d'ordre effectue un rangement des éléments de $\mathscr{E}$, sans qu'il n'y ait d'*ex aequo* possible[3]. Nous noterons $>$ la relation d'ordre strict correspondant à la relation $\geq$, c'est-à-dire la relation telle que $x > y \Leftrightarrow x \geq y$ et non $y \geq x$. De même, $\max_{\geq} \mathscr{E}$ désignera l'ensemble des éléments de $\mathscr{E}$ non dominés (optimaux) pour $\geq$, c'est-à-dire l'ensemble $\mathscr{S} \subseteq \mathscr{E}$ tel que pour tout $\widehat{x} \in \mathscr{S}$ on a $\widehat{x} \not> y$ pour tout $y \in \mathscr{E}$.

Le graphe représentant une relation d'ordre ne comporte pas d'arc non-orienté (propriété d'antisymétrie) autre que les arcs (x, x) que l'on omet. On pourra, pour simplifier la représentation, omettre les arcs obtenus par transitivité (voir figure 1.2).

Définition 1.8 (Relation de préordre) *Une* relation de préordre $\succeq$ *est une relation binaire réflexive et transitive.*

Une relation de préordre effectue aussi un rangement des éléments de $\mathscr{E}$, mais cette fois-ci les *ex aequo* sont possibles. On définit de la même façon que pour une relation d'ordre la relation de préordre strict $\succ$, ainsi que l'ensemble des éléments non dominés $\max_{\succeq} \mathscr{E}$.

[3]En revanche il peut y avoir des incomparabilités si la relation est incomplète.

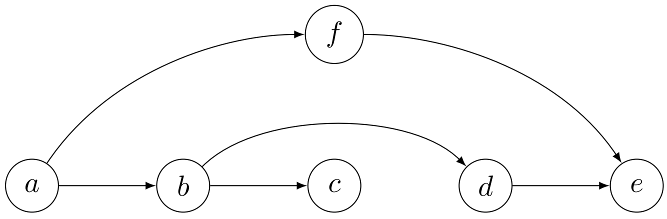


Figure 1.2 — Représentation d'un ordre non complet sous forme d'un graphe : l'ordre représenté est la clôture transitive de $a \geq b \geq c$, $b \geq d \geq e$, $a \geq f \geq e$.

Le graphe représentant une telle relation est constitué de groupes de sommets (*clusters*) formant des cliques non orientées, liés entre eux ou non par des arcs orientés. Pour simplifier la notation, nous représenterons une telle relation binaire de la même manière qu'un ordre, mais en faisant figurer dans les sommets du graphe l'ensemble des éléments de la même classe d'équivalence, ou un représentant de la classe d'équivalence (voir figure 1.3).

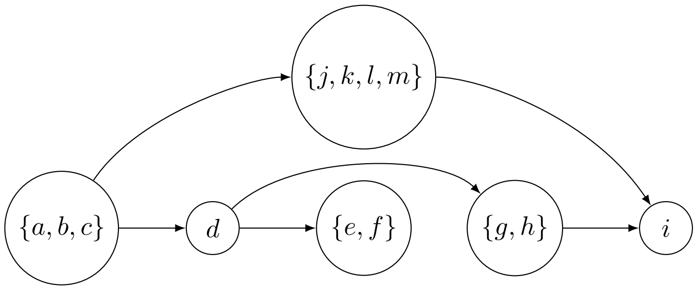


Figure 1.3 — Représentation d'un préordre non complet sous forme d'un graphe : l'ordre représenté est la clôture transitive de $\{a, b, c\} \geq d \geq \{e, f\}$, $d \geq \{g, h\} \geq i$, $\{a, b, c\} \geq \{j, k, l, m\} \geq i$.

1.2.1.2 Structure de préférence ordinale

Le modèle classique (voir [Vincke, 1989] par exemple) de représentation des préférences en théorie de la décision est fondé sur la question suivante : «Étant données deux alternatives x et y, x est-elle au moins aussi bonne que y ?». Répondre à cette question par *oui* ou *non* de manière non ambiguë pour toute paire d'éléments (x, y) revient à définir une relation binaire (supposée, par définition, réflexive) sur l'ensemble des alternatives :

Définition 1.9.a (Structure de préférence ordinale) *Soit $\mathscr{E}$ un ensemble d'alternatives. Une* structure de préférences ordinale *sur $\mathscr{E}$ est une relation binaire réflexive notée $\Re_S$.*

Pour toute paire d'alternatives (x, y) et pour toute structure de préférences ordinale $\Re_S$, nous pouvons être confrontés aux trois situations mutuellement exclusives suivantes :

- ▷ **indifférence :** $x \Re_S y$ et $y \Re_S x$ s'interprète comme «x est indifférent à y» ;
- ▷ **incomparabilité :** non $x \Re_S y$ et non $y \Re_S x$ s'interprète comme «x est incomparable à y» ; il peut s'agir d'un refus de comparer (point de vue éthique) ou d'une incapacité de comparer due à un manque de connaissances (point de vue épistémique) ;
- ▷ **préférence stricte :** $x \Re_S y$ et non $y \Re_S x$ (respectivement $y \Re_S x$ et non $x \Re_S y$) s'interprète comme «x (resp. y) est strictement préféré à y (resp. x)».

Ces trois relations nous fournissent une définition alternative (mais fondamentalement équivalente) pour la notion de structure de préférence ordinale :

Définition 1.9.b (Structure de préférence ordinale) *Soit $\mathscr{E}$ un ensemble d'alternatives. Une* structure de préférences ordinale *sur $\mathscr{E}$ est un triplet $(\Re_P, \Re_I, \Re_R)$ de relations binaires vérifiant :*
- ▷ *$\Re_P$ est asymétrique ;*
- ▷ *$\Re_I$ (relation d'indifférence) est réflexive et symétrique ;*
- ▷ *$\Re_R$ (relation d'incomparabilité) est irréflexive et symétrique ;*
- ▷ *pour toute paire d'alternatives (x, y), on a $x\Re_P y$ ou $y\Re_P x$ ou $x\Re_I y$ ou $x\Re_R y$, cette disjonction étant exclusive.*

On peut obtenir la relation $\Re_S$ à partir du triplet $(\Re_P, \Re_I, \Re_R)$ en posant : $\Re_S = \Re_P \cup \Re_I$. Réciproquement, on a, pour tout couple d'alternatives (x, y) :
- ▷ $x\Re_P y \Leftrightarrow x\Re_S y$ et non $y\Re_S x$ (x est strictement préféré à y si et seulement si x est aussi bon que y mais y n'est pas aussi bon que x),
- ▷ $x\Re_I y \Leftrightarrow x\Re_S y$ et $y\Re_S x$ (x est indifférent à y si et seulement si x est aussi bon que y et y est aussi bon que x),
- ▷ $x\Re_R y \Leftrightarrow$ non $x\Re_S y$ et non $y\Re_S x$ (x et y sont incomparables si et seulement si on ne peut ni dire que x soit aussi bon que y, ni que y soit aussi bon que x).

Un exemple de structure de préférence ordinale «dégénérée» utilisée en intelligence artificielle est la structure de préférence dichotomique. Dans une telle structure, les préférences ne sont données que par un ensemble de bonnes alternatives : toute alternative appartenant à cet ensemble est meilleure que toute alternative n'appartenant pas à cet ensemble, mais les bonnes alternatives sont indifférentes entre elles (de même que les mauvaises).

Définition 1.10 (Structure de préférence dichotomique) *Une* structure de préférence dichotomique *sur un ensemble d'alternatives $\mathscr{E}$ est une structure de préférence ordinale particulière définie par la donnée d'un sous-ensemble d'alternatives $\mathscr{G} \subset \mathscr{E}$. La relation $\Re_S$ est un préordre total défini comme suit : $\forall (x, y) \in \mathscr{E}^2$, $x\Re_S y \Leftrightarrow x \in \mathscr{G}$ ou $y \in \mathscr{E} \setminus \mathscr{G}$.*

En d'autres termes, une structure de préférence dichotomique s'intéresse à des préférences pouvant être représentées par un préordre total dont la relation d'équivalence associée possède deux classes d'équivalence. La raison de l'intérêt de cette structure de préférence en intelligence artificielle est qu'elle concentre, malgré son aspect assez fruste, toute la complexité computationnelle de langages liés à des modèles de préférences plus évolués. Nous aurons l'occasion de revenir sur ce point dans le chapitre 3 consacré aux langages de représentation compacte.

1.2.1.3 Extensions de la structure de préférence ordinale

Dans le modèle des structures de préférences, on n'autorise que les réponses *oui* ou *non* à la question «x est-il au moins aussi bon que y ?». Cependant, on pourrait vouloir apporter des précisions à cette réponse, par exemple en incluant des informations sur l'*intensité* de la préférence, ou encore des informations sur la *crédibilité* de la proposition «x est préféré à y», ou modéliser des situations d'hésitation (voir par exemple un modèle de préférences un peu plus général présenté dans [Roy, 1985]). De nombreuses structures de préférences ont été étudiées dans la littérature. Certaines permettent de prendre en compte des seuils d'indifférence, d'autres acceptent l'incomparabilité entre alternatives, d'autres incluent la notion d'incertitude ou d'imprécision.

La première extension classique est la structure de préférence qualitative, qui ajoute une idée d'intensité aux préférences :

Définition 1.11 (Structure de préférence qualitative) *Une structure de préférence* qualitative *sur $\mathscr{E}$ est un couple $(\langle \mathscr{V}, \succeq \rangle, u)$, où $\langle \mathscr{V}, \succeq \rangle$ est une structure de valuation quantitative formée d'un ensemble $\mathscr{V}$ totalement ordonné par $\succeq$, et u est une fonction d'utilité de $\mathscr{E}$ dans $\mathscr{V}$.*

En d'autres termes, une structure de préférence qualitative associe à chaque alternative une valuation. Aucune hypothèse spécifique n'est requise sur l'espace de valuation, mis à part le fait qu'il soit totalement ordonné. Par exemple, cet espace peut être $\mathscr{V} = \{$médiocre, mauvais, moyen, bon, excellent$\}$; un tel espace pourra être muni de la relation d'ordre suivante $\geq$ définie comme suit : médiocre $\leq$ mauvais $\leq$ moyen $\leq$ bon $\leq$ excellent.

Le modèle de représentation des préférences introduit avec la structure de préférence qualitative, s'il est plus riche que la structure de préférence ordinale, est cependant trop pauvre pour effectuer des comparaisons entre les intensités. Pour pallier ce manque d'expressivité, la plupart des travaux s'intéressent à des structures de valuation numériques, qui possèdent une loi de composition interne, permettant notamment d'ajouter les valuations, et surtout de faire leur différence.

Définition 1.12 (Structure de préférence cardinale) *Une structure de préférence* cardinale *sur $\mathscr{E}$ est un couple $(\langle \mathscr{V}, \succeq, \oplus \rangle, u)$, où $(\langle \mathscr{V}, \succeq \rangle, u)$ est une structure de préférence qualitative, et $\oplus$ est une loi de composition interne associative et commutative sur $\mathscr{V}$ ayant les propriétés suivantes :*
- ▷ *monotonie : $\forall a, b, c \in \mathscr{V}$ tels que $a \preceq c$, on a $(a \oplus b) \preceq (c \oplus b)$,*
- ▷ *élément neutre : $\forall a \in \mathscr{V}$, $a \oplus \bot = a$,*
- ▷ *élément absorbant : $\forall a \in E$, $a \oplus \top = \top$,*
- ▷ *existence d'une différence unique : Soit $\alpha \preceq \beta$ alors $\max\{\gamma \mid \alpha \oplus \gamma = \beta\}$ existe et est noté $\beta \ominus \alpha$.*

En d'autres termes, $\oplus$ est une *conorme* pour laquelle la notion de différence est définie, et $\langle \mathscr{V}, \succeq, \oplus \rangle$ est un monoïde commutatif totalement ordonné. Classiquement, on choisit des utilités à valeurs réelles (soit $\mathscr{V} = \overline{\mathbb{R}}$ ou $\mathscr{V} = \overline{\mathbb{R}}^+$) ou à valeurs entières ($\mathscr{V} = \overline{\mathbb{N}}$), ces espaces de valuation étant munis de l'ordre naturel $\geq$ sur les nombres, et de la loi $+$.

Dans cette définition la fonction u est appelée *fonction d'utilité*. Cette définition de fonction d'utilité est plus générale que celle qui est classiquement introduite dans les ouvrages traitant de la théorie du *welfarisme*[4], qui ne considèrent que des fonctions d'utilités définies sur $\mathbb{R}$ ou $\mathbb{R}^+$. Notre point de vue s'inspire davantage des formalismes introduits en intelligence artificielle pour la modélisation des préférences au sens large, par exemple dans les problèmes de satisfaction de contraintes valués [Cooper et Schiex, 2004] (notion de structure de valuation *juste*), le cadre PFU [Pralet, 2006], ou les problèmes de partage [Fargier *et al.*, 2004a]. Nous reparlerons de la théorie de l'utilitarisme un peu plus loin dans ce chapitre, et des problèmes de satisfaction de contraintes dans le chapitre 5 consacré à l'algorithmique.

Une question classique posée en théorie de la décision est la question de la représentativité d'une telle structure de préférence en terme de structure de préférence ordinale. En d'autres termes, on cherche quel type d'ordre est représentable par une structure de préférence cardinale. Nous avons la proposition suivante :

Proposition 1.1 *Soient $\mathscr{E}$ un ensemble d'alternatives fini ou infini dénombrable, et $(\langle \mathbb{R}, \succeq, + \rangle, u)$ une structure de préférence cardinale sur $\mathscr{E}$. Alors il existe une structure de préférence ordinale $\Re_S$ telle que pour tout $(x, y) \in \mathscr{E}^2$, $x \Re_S y \Leftrightarrow u(x) \succeq u(y)$ si et seulement si $\Re_S$ est un préordre total.*

Si de plus on a $u(x) = u(y) \Leftrightarrow x = y$, alors $\Re_S$ est un ordre total.

On dira que u représente *la relation $\Re_S$.*

Cette proposition, classique en théorie de la décision [Vincke, 1978, 1989; Bouyssou et Vincke, 2006] a été démontrée notamment par Cantor [Cantor, 1915]. Elle permet d'affirmer notamment

[4] Il ne semble pas exister d'équivalent français pour ce terme, dont le sens diffère légèrement de celui de l'utilitarisme. Nous prendrons donc le risque de troubler les puristes de la langue française et emploierons cet anglicisme francisé.

l'existence d'une fonction d'utilité à valeurs réelles pour toute structure de préférence de type préordre ou ordre totaux. Bien entendu, cette représentation numérique n'est pas unique, puisqu'elle est définie à une transformation croissante près. Le problème de construction d'une fonction d'utilité à partir d'une relation de préordre est le problème de *représentation numérique*, classique dans le domaine de la modélisation des préférences.

D'autres modèles plus riches que la structure de préférences cardinale ont été introduits. On citera par exemple :

- ▷ les modèles à seuil, fondés sur une fonction $g : \mathscr{E} \to \mathbb{R}$ et une constante $q \geq 0$ appelée «seuil d'indifférence» ;
- ▷ les modèles à base d'intervalles, fondés sur deux fonctions $g : \mathscr{E} \to \mathbb{R}$ et $q : \mathbb{R} \to \mathbb{R}^+$.

Ces modèles sont nés de l'inadéquation du modèle du préordre à certaines situations courantes pour lesquelles la relation d'indifférence n'est pas forcément transitive. Cette hypothèse de transitivité est criticable dans des contextes où l'on est incapable de discriminer des alternatives proches, mais où l'on sait discriminer des alternatives plus éloignées (l'introduction de cette remarque dans le contexte de la modélisation des préférences est due à [Luce, 1956]). L'exemple classique associé à cette situation est celui du sucre dans une tasse de café. Si l'on désigne par T_i une tasse de café contenant i milligrammes de sucre, il est très probable qu'un agent appréciant un café très sucré ne sera pas indifférent entre T_n et T_o, pour n assez grand, mais ne sera pas à même de faire la différence entre T_i et T_{i+1}. Cela suggère une relation d'indifférence intransitive (car $\forall i \; T_i \Re_I T_{i+1}$, mais non $T_n \Re T_0$).

Ces modèles permettent respectivement de représenter des préférences à base :

- ▷ de **semi-ordres** (ou quasi-ordres) $\Re_S : \forall (x, y) \in \mathscr{E}^2,\ x \Re_S y \Leftrightarrow g(x) \preceq g(y) + q$ (voir par exemple [Vincke, 1978]) ;
- ▷ d'**ordres d'intervalle** $\Re_S : \forall (x, y) \in \mathscr{E}^2,\ x \Re_S y \Leftrightarrow g(x) \preceq g(y) + q(g(y))$ (voir par exemple [Pirlot et Vincke, 1997]).

Nous pouvons noter l'extension récente de ces modèles à base d'intervalles aux intervalles à 3 points (voir par exemple [Öztürk et Tsoukiàs, 2006]) qui permettent aussi de prendre en compte une intransitivité de la relation d'indifférence, mais en utilisant uniquement des informations ordinales (la distance entre les points n'est pas importante).

Nous ne détaillerons pas plus ici l'ensemble de ces structures de semi-ordres ou d'ordres d'intervalle. Nous invitons le lecteur à consulter les quelques références citées pour plus de détails sur le sujet.

Notons enfin l'existence d'un autre type de structure de préférence classique en intelligence artificielle, qui raffine la structure de préférence cardinale : la structure de préférence *floue* [Fodor et Roubens, 1994; Perny et Roy, 1992]. Une telle structure est fondée sur une fonction de *crédibilité* μ, qui à toute paire d'alternatives (x, y) associe un nombre $\mu(x, y) \in [0, 1]$, représentant la crédibilité de l'assertion «x est au moins aussi bon que y ».

Concluons cette section sur la modélisation des préférences par un petit résumé des structures de préférence classiques introduites :

- ▷ Une structure de préférence *dichotomique* est un sous ensemble $\mathscr{G} \subseteq \mathscr{E}$.
- ▷ Une structure de préférence *ordinale* est une relation binaire réflexive $\Re_S$.
- ▷ Une structure de préférence *qualitative* est fondée sur une fonction d'utilité u qui à chaque alternative x associe une valuation $u(x) \in \mathscr{V}$. Les valuations sont comparées grâce à une relation d'ordre sur $\mathscr{V}$.
- ▷ Une structure de préférence *cardinale* est une structure de préférence qualitative particulière, dans laquelle $\mathscr{V}$ est un monoïde commutatif totalement ordonné.
- ▷ Une structure de préférence *à seuil* est fondée sur une fonction $g : \mathscr{E} \to \mathbb{R}$ et un seuil

d'indifférence $q \geq 0$. Les alternatives x sont considérées comme indifférentes si la différence entre leurs valuations sont inférieures au seuil q.

- Une structure de préférence *à base d'intervalles* associe à chaque alternative x un intervalle $[g(x), g(x) + q(g(x))]$. Il s'agit d'une structure de préférence à seuil variable.
- Une structure de préférence *floue* sur $\mathscr{E}$ est fondée sur une fonction de crédibilité $\mu : \mathscr{E} \times \mathscr{E} \rightarrow [0, 1]$, qui à toute paire d'alternatives (x, y) associe une mesure de crédibilité $\mu(x, y)$.

1.2.2 L'espace cible des préférences individuelles

Dans de nombreux problèmes de décision, l'espace cible des préférences $\mathscr{E}$ apparaît de manière naturelle. Il s'agit de l'ensemble des alternatives entre lesquelles le choix s'effectue, ou en d'autres termes, de l'ensemble des actions possibles de l'agent décideur. Dans les problèmes de partage, si l'espace cible des préférences collectives est naturellement l'ensemble des allocations[5], on fait souvent une hypothèse simplificatrice en ce qui concerne les préférences individuelles : on suppose dans la plupart des problèmes que la satisfaction d'un agent n'est pas du tout affectée par la part reçue par les autres agents. En d'autres termes, on suppose que l'espace cible des préférences des agents est simplement l'ensemble des parts possibles $\wp(\mathscr{R})$.

De manière générale, on distinguera deux cas de figure :

- le cas de préférences individuelles non exogènes, dont l'espace cible est simplement l'ensemble des parts, et donc la satisfaction des agents ne dépend pas de la part des autres agents ;
- le cas de préférences individuelles exogènes, dont l'espace cible est maintenant l'ensemble des partages possibles, et donc la satisfaction des agents dépend de l'ensemble du partage.

On peut classer dans la seconde catégorie l'ensemble des partages qui font intervenir la notion de jalousie ou d'envie directement dans les préférences des agents. Cette modélisation reflète de manière plus réaliste le comportement humain vis-à-vis de la décision collective que ne le fait le modèle rationnel de l'*Homo Œconomicus*. On pourra lire avec profit [Henrich *et al.*, 2001; Zizzo et Oswald, 2000], cités par [Delahaye, 2005], sur le sujet de la non rationalité des préférences humaines : les expériences conduites par les auteurs de ces articles révèlent notamment que dans certains cas, un agent préfère diminuer sa satisfaction personnelle afin de nuire à un agent plus chanceux que lui.

Notons malgré tout que l'hypothèse de non exogénéité des préférences individuelles est une approximation satisfaisante dans la plupart des problèmes concrets étudiés. La notion de jalousie et d'envie pourra, dans ce cadre-là, se traduire par une propriété exigée sur le partage résultant, et non comme une propriété intrinsèque des préférences individuelles. Nous reviendrons sur le sujet dans la section 1.3.1.3 consacrée à l'absence d'envie.

Notons que, lorsque l'espace cible des préférences individuelles est l'ensemble des parts possibles, nous pouvons définir une propriété supplémentaire sur la structure de préférences, la propriété de monotonie :

Définition 1.13 (Monotonie) *Soit $\succeq$ une structure de préférences ordinale sur l'espace des parts possibles $\wp(\mathscr{R})$. Alors $\succeq$ est* monotone *si et seulement si pour tout couple de parts $(\pi, \pi') \in \wp(\mathscr{R}) \times \wp(\mathscr{R})$, $\pi \subseteq \pi' \Rightarrow \pi \preceq \pi'$.*

La propriété de monotonie est intuitive et raisonnable. Pour un agent ayant des préférences monotones, l'ajout d'un peu de ressource (un objet par exemple) dans la part qu'il a ne pourra pas avoir d'effet «négatif». Bien entendu, cela n'est pas toujours le cas (voir par exemple 3.9 dans le chapitre 3), mais on pourra considérer pour les problèmes de partage auxquels on aura affaire que

[5]On peut éventuellement restreindre la définition des préférences collectives à l'ensemble de partages admissibles sans que cela n'ait d'incidence sur la suite.

c'est le cas en première approximation. Si un agent ne veut pas d'un objet, il pourra toujours le donner à un autre agent ou tout simplement le jeter ou l'oublier dans un coin de grenier.

1.3 Agrégation des préférences et partage équitable

Nous avons jusqu'ici posé les bases nécessaires à la modélisation du problème de partage de biens indivisibles. Nous nous retrouvons maintenant face à un problème délicat, qui constitue le cœur du partage : celui de l'agrégation des préférences individuelles. Ce problème peut être posé informellement de la manière suivante :

> Étant donnés un ensemble d'objets et un ensemble d'agents ayant des préférences sur les parts possibles qu'ils peuvent recevoir, comment partager les objets entre les agents, de manière à ce que le partage soit le plus équitable possible ?

Ce problème de *justice distributive*, vieux comme le monde, a été abondamment étudié par les philosophes et les économistes, car il est lié au développement de toute société : du partage de territoires de chasse dans les sociétés primitives au partage des zones d'exploitations minières mondiales, la problématique de la répartition de biens communs est au cœur des interactions et activités collectives humaines. Indissociable de la notion de partage, le concept d'*équité* a été également abondamment étudié dans de nombreux domaines. Comme nous l'avons rappelé en introduction, la notion d'équité ne véhicule pas nécessairement une idée d'éthique ou de morale, contrairement à ce qui semble être d'usage dans le langage courant. Nous nous conformerons à l'acception de [Young, 1994] :

> «By "equitable" I do not necessarily mean ethical or moral, but that which a given society considers to be *appropriate* to the need, status, and contribution of its various members.»

Le problème de justice distributive est historiquement celui des philosophes et des économistes théoriciens. Les premiers ont concentré leur attention sur la signification de concepts aussi abstraits que l'équité ou la justice, alors que les seconds (ainsi que quelques mathématiciens) se sont intéressés à la modélisation et à l'axiomatisation de ces concepts et de propriétés qui leur sont liées. En revanche, peu d'entre eux se sont intéressés à des propriétés liées à la construction ou à l'existence de partages équitables dans certaines conditions bien précises [Brams et Taylor, 1996]. Ce domaine, plus récent, est plutôt l'apanage des sciences politiques, de la sociologie, ou de l'économie appliquée, qui requièrent des approches plus empiriques. Ces travaux ont ouvert la voie à l'extension du domaine du partage — et plus généralement du choix social — à des sciences comme l'informatique ou l'intelligence artificielle, s'intéressant principalement à des aspects liés à la représentation compacte, à l'algorithmique ou à la complexité des problèmes de choix social.

Nous allons présenter ici en quelques pages les fondements théoriques principaux qui sont à la base de la modélisation du problème de partage en économie.

1.3.1 Principes normatifs de la justice distributive

1.3.1.1 Le principe d'équité

Il existe trois grandes théories normatives de la justice distributive. La première d'entre elles, et la plus ancienne, est le principe d'équité d'Aristote :

> «Les contestations et les plaintes naissent quand, étant égales, les personnes possèdent ou se voient attribuer des parts non égales, ou quand, les personnes n'étant pas égales, leurs parts sont égales. [...] Tous les hommes reconnaissent, en effet, que la justice dans

la distribution doit se baser sur un mérite de quelque sorte, bien que tous ne désignent pas le même mérite.»

Aristote, *Éthique à Nicomaque*, Livre V, chapitre 6, traduction Tricot.

Le principe selon lequel les égaux doivent être traités de manière égale prête relativement peu à confusion : si deux personnes sont parfaitement identiques selon toutes les caractéristiques entrant en ligne de compte dans le problème, alors elles doivent être traitées de manière parfaitement égales. En revanche, le principe de traitement inégal des inégaux — de manière proportionnelle à leurs différences — est sujet à de nombreuses interprétations.

Derrière le principe d'équité d'Aristote se cache quatre définitions de la «pertinence» des critères, au cœur de toutes les réflexions d'ordre philosophiques sur la justice distributive [Moulin, 2003].

1. Le principe de *compensation*. L'idée qui est à la base de ce principe est que certains agents ont besoin d'une plus grande quantité de ressources basiques que d'autres agents pour atteindre le même degré de bien-être, et ce à cause d'un certain nombre de différences involontaires et moralement injustifiées (santé, richesse des parents, capacités intellectuelles, ...). Le principe de compensation suggère de donner plus de ressource aux personnes qui en ont le plus besoin : en d'autres termes, on cherche à atteindre l'égalité *ex-post*.
2. Le principe de *récompense*. Dans certains cas, les différences sur les caractéristiques individuelles sont volontaires : les agents peuvent en être tenus pour responsables. Dans ces cas-là, ces différences justifient un traitement inégal des agents et doivent être prises en compte lors de la division de la ressource. Selon ce principe, l'attribution de la ressource se fait en vertu du mérite des agents : plus un individu a contribué à la création de la ressource, plus il doit en bénéficier.
3. Le principe de *droits exogènes*. Certains principes guidant l'allocation de la ressource viennent de considérations complètement extérieures à la consommation de cette ressource et des questions du type *qui en a besoin ?* et *qui la mérite ?* qui lui sont rattachées. L'illustration la plus édifiante de ce principe est le principe d'égalité dans l'allocation des droits politiques : chaque citoyen ayant atteint la majorité a le droit de voter, quels que soient ses mérites, son niveau d'étude ou encore son intérêt pour la politique. Ce principe est celui de l'(in)égalité *ex-ante* (nous reviendrons sur les droits exogènes au chapitre 2).
4. Le principe d'*adéquation* (ou *fitness*). Ce principe peu être résumé en une phrase : *La ressource doit être donnée à la personne qui en fait le meilleur usage.* Il se décline en deux principes : l'adéquation à la somme et l'adéquation à l'efficacité, qui correspondent respectivement à l'utilitarisme classique et au principe d'efficacité de Pareto, dont nous parlerons plus loin.

Outre la critique concernant la difficulté de juger des inégalités entre les individus (comment en effet juger de critères aussi flous que «la contribution de chacun», ou encore «le bon usage de la ressource»), on oppose en général au principe d'Aristote le fait qu'il ne fonctionne parfaitement que si la ressource à partager est divisible [Young, 1994].

1.3.1.2 Le *welfarisme* cardinal

Le *welfarisme* compte parmi les paradigmes dominants actuellement dans le domaine de la micro-économie. Née des travaux initiaux des précurseurs Condorcet et Borda, puis de ceux de Bentham et d'Arrow, cette théorie s'applique de manière générale à tous les problèmes de décision collective (dont les problèmes de partage sont des instances particulières). Elle est fondée sur un postulat de choix rationnel — chaque choix individuel cherche à maximiser une relation de préférence donnée complète —, et sur le principe de l'individualisme méthodologique : l'individu et le monde extérieur (caractérisé par un ensemble d'états, ou en d'autres termes d'alternatives) sont deux entités clairement séparées. L'autorité collective peut agir sur la distribution des ressources, mais pas sur l'individu lui-même, qui a des caractéristiques intrinsèques telles que ses valeurs, préférences, expériences, etc. Le postulat de base de la théorie du *welfarisme* est que chaque agent peut exprimer sa satisfaction vis-à-vis des états du monde sous la forme d'un ordre sur ces états ou d'un indice numérique : le *bien-être social* (*social welfare*). Le *welfarisme* est donc un procédé permettant d'agréger de manière mécanique le bien-être social des agents pour en déduire une décision collective. Ce modèle se divise en deux grands domaines d'étude :

- ▹ le *welfarisme* ordinal, ou *choix social*, qui s'applique à l'agrégation de relations de préférences ordinales, comme dans le domaine du vote ;
- ▹ le *welfarisme* cardinal, version quantitative du problème de décision collective, qui axiomatise le principe de l'utilitarisme de Bentham.

On pourra consulter par exemple l'ouvrage de référence [Arrow *et al.*, 2002] afin d'avoir une synthèse détaillée de la théorie du choix social, s'étendant de l'agrégation des préférences ordinales à la théorie de l'utilitarisme.

Utilitarisme classique et égalitarisme Le *welfarisme* cardinal, et *a fortiori* le domaine de la justice distributive, s'appuient historiquement sur deux théories d'importance en économie. La première de ces théories, introduite principalement par Jeremy Bentham (1748–1832) et John Stuart Mill (1806–1873) sous sa forme systématique est celle de l'utilitarisme classique. L'idée fondatrice de cette théorie est qu'il est possible de représenter la satisfaction d'un agent vis-à-vis d'un état par un indice numérique, qui représente formellement la somme des joies et des peines de l'individu en question : l'utilité doit être comprise comme une mesure de satisfaction psychique cardinale qui peut être ajoutée entre les individus. Les biens doivent être répartis de manière à maximiser le bien-être social total des demandeurs (le meilleur bien pour le plus grand nombre). De manière plus formelle, l'utilitarisme cherche à maximiser la somme des utilités individuelles : les incréments d'utilité individuelle de différents agents sont complètement interchangeables. Pour comprendre ce point de vue, il suffit de considérer chaque agent comme un producteur de bien-être social : le but est de maximiser la production totale de bien-être social, sans se préoccuper des inégalités entre les agents.

Deux principales critiques ont été opposées à cette théorie. Tout d'abord, le fait que l'on puisse comparer entre des individus des niveaux de satisfaction correspondant à des états psychiques internes est plus que discutable. La seconde critique concerne le fait que cette théorie peut exiger le sacrifice de quelques-uns pour le bonheur du plus grand nombre : ce principe moral n'est pas universellement accepté.

Ces objections à la théorie de l'utilitarisme classique de Bentham ont donné naissance à la théorie de l'égalitarisme de Rawls [Rawls, 1971]. L'idée fondatrice de cette théorie est qu'une distribution est équitable si le plus malheureux des individus est rendu le plus heureux possible. Pour l'égalitariste pur, les compensations entre agents sont impossibles ; un gain très important d'utilité pour tous les agents sauf un ne compense pas une perte minuscule d'utilité pour ce dernier agent s'il s'avère qu'il est déjà le moins satisfait. Notons que cette idée n'implique pas nécessairement une égalisation des revenus entre les agents, car d'un point de vue économique une telle égalisation n'incite pas à la création de richesse et donc conduit à la diminution de la quantité de biens disponible. Contrairement aux apparences, cette théorie n'est pas une théorie utilitariste dans le sens strict du terme, et en cela, elle répond à la première des deux objections concernant l'utilitarisme classique. En effet, si le niveau de «bonheur» est encore ici mesuré par un indice numérique, cet indice ne fait pas référence à un état psychique interne, mais à des moyens par lesquels on peut assurer le bonheur (revenu, santé, etc.) : les biens primaires.

Les principales critiques opposées à cette théorie sont les suivantes. Tout d'abord, même si le recours à des biens primaires pour la mesure du bien-être d'un agent permet de pallier le problème d'intercomparabilité des préférences, l'introduction de ces biens pose d'autres problèmes : absence de comparaison objective pour certains biens primaires (respect par exemple), ou encore difficulté de déterminer un niveau d'importance relative entre ces biens. En d'autres termes, l'introduction de biens primaires ne résoud pas le problème de comparaison interpersonnelle des préférences, mais ne fait que le reporter un peu plus loin. L'autre critique classique est liée à la définition-même du critère égalitariste : est-il juste d'imposer des restrictions sévères à la grande majorité des individus d'une société afin d'augmenter de manière infime les revenus de l'individu le plus pauvre ?

Bien que le débat entre utilitaristes et égalitariste soit ancien, il s'est illustré vers le milieu du XX^{e} siècle par celui entre deux philosophes sociaux, Rawls [Rawls, 1971], plaidant pour l'égalitarisme, et Harsanyi [Harsanyi, 1955], argumentant en faveur de l'utilitarisme. Les deux visions des choses correspondent à deux interprétations différentes de la *loi d'ignorance* (*Rawlsian veil of ignorance*) : « Si un individu devait rejoindre une société en ignorant tout de la place qu'il occuperait dans cette société, quel principe de décision collective jugerait-il juste pour cette société ?» Là où les égalitaristes considèrent que l'individu a une aversion pour le risque, et craint de se retrouver à la place du plus pauvre, l'individu utilitariste a une attitude bayésienne et cherche à maximiser son utilité espérée.

Macro- contre micro-*welfarisme* Les critiques opposées à la conception utilitariste de la justice distributive sont justifiées dans un contexte macro-*welfariste* [Sen, 1992] : l'idée de représenter la somme des joies et des peines d'un individu (même si l'on passe par l'intermédiaire de biens primaires) est plus que discutable si l'on raisonne de manière globale, pour les raisons que nous avons évoquées ci-avant. En revanche, cette théorie est acceptable dans un contexte micro-économique (micro-*welfariste*), dans le sens où l'on établit une franche séparation entre le problème en cours d'étude et le reste des caractéristiques de l'agent ainsi que les autres individus non concernés. Dans le contexte de problèmes de «micro-allocation», l'interprétation de l'utilité d'un agent ne concerne que le problème en cours, et donc ne fait pas référence à un niveau de contentement global de l'individu.

Ainsi, la théorie du *welfarisme* constitue un outil formel remarquable pour traiter les problèmes de justice distributive localisés que sont en général les problèmes de partage. Appliqué aux problèmes de partage, ce modèle permet d'explorer tout un ensemble de compromis entre le principe de compensation (invoqué par les égalitaristes), et le principe d'adéquation (à la base de l'utilitarisme classique). La question de la manière d'y intégrer le principe de *droits exogènes* sera abordée au chapitre 2. En revanche, le *welfarisme* est complètement inadapté pour l'ensemble des problèmes faisant intervenir le principe de récompense. On peut citer parmi ceux-là les problèmes de partage de coûts ou de surplus entre des agents, concernant les problèmes impliquant des agents ayant contribué à hauteur inégale dans la ressource, sous la forme d'un investissement initial inégal par exemple [Moulin, 2002]. Ces problèmes peuvent être traités à l'aide de modèles tels que celui de la *valeur de Shapley* [Shapley, 1953], introduits initialement dans le contexte de la théorie des jeux de von Neumann et Morgenstern, impliquant la notion de répartition de gains entre plusieurs membres d'une même coalition. Ces modèles permettent de prendre en compte de manière générale le principe de récompense, ou de mérite.

Notons enfin que la théorie du *welfarisme* est une théorie de la justice résultat, ou justice *téléologique* (*endstate justice*). En d'autres termes, on cherche à assurer l'équité du résultat du partage, sans vraiment se soucier de la manière dont ce résultat peut être obtenu[6]. Cette vision

[6] C'est par exemple le principe sur lequel est fondée la notion de discrimination positive.

est inadéquate pour certains types de problèmes de partage pour lesquels l'équité ne peut pas être obtenue. Citons par exemple le cas d'allocation de reins disponibles pour un ensemble de patients en attente de greffe. Le *welfarisme* s'avère impuissant à résoudre ce type de problèmes, car ils nécessitent une équité dans la *procédure* d'allocation, et non dans le résultat final : il ne peuvent être traités par les principes de la justice téléologique, mais ils relèvent de la justice *procédurale*. Les modèles concernant la justice procédurale, fondés sur des notions éthiques et morales liées au hasard, à la priorité ou à l'équité par rotation (lorsque le bien à partager le permet), diffèrent de ceux concernant la justice téléologique, et jusqu'à ce jour aucun modèle n'englobe ces deux aspects très différents de la justice distributive.

1.3.1.3 L'absence d'envie

Les objections philosophiques et conceptuelles opposées aux modèles de l'utilitarisme classique et de l'égalitarisme ont conduit certains économistes à adopter un point de vue entièrement différent. L'un des écueils de ces approches est la comparaison interpersonnelle des utilités : le décideur (qui peut être une entité abstraite représentant la collectivité) doit être capable de comparer lui-même les utilités d'individus dont il ne sait rien par ailleurs. L'idée à la base de l'approche fondée sur l'absence d'envie est que ce sont les agents eux-mêmes qui jugent si leur situation est meilleure que celle des autres. En d'autres termes, un partage est sans-envie si aucun agent estime qu'il est moins heureux avec sa part qu'il ne le serait avec la part d'un autre agent, *selon son propre point de vue*.

La notion d'absence d'envie apparaît pour la première fois dans [Tinbergen, 1953], qui introduit un critère d'équité d'une société fondé sur la notion d'envie. Du point de vue de Tinbergen, une société est équitable si et seulement si aucun des individus qui la composent ne désire échanger sa place avec quelqu'un d'autre : il s'agit d'une vision *forte* de l'absence d'envie. Cette propriété, impossible a obtenir dans le cas général, car elle porte sur tous les critères confondus, a été introduite dans un sens plus faible dans [Foley, 1967]. Dans ce contexte, elle n'est pas appliquée au sens global, mais seulement pour un problème d'allocation particulier, pour lequel un agent compare uniquement sa part avec celle des autres. La différence entre absence d'envie forte et absence d'envie faible est bien entendu très similaire à la distinction entre vision macro-*welfariste* et micro-*welfariste*

La propriété d'absence d'envie est séduisante, car c'est une notion purement ordinale, et qui ne requiert aucune élicitation des préférences des agents sur une échelle numérique commune. En revanche, elle ne s'applique pas toujours. Tout d'abord, elle est incompatible avec toute idée de mérite, contribution, besoin, et plus généralement, elle est incompatible avec toute idée de jugement de valeur, car un tel jugement de valeur est toujours fondé sur la comparaison interpersonnelle des préférences. En outre, ce critère peut être simplement inadapté à un problème pour des raisons plus «mécaniques» que philosophiques :

- ▷ Les préférences des agents peuvent être complètement disjointes. Considérons par exemple l'application 1 concernant la constellation de satellites Pléiades. Dans ce problème, les demandes de chaque agent concernent des zones géographiques différentes. On constate donc *a priori* qu'il ne peut y avoir aucune envie dans ce problème. En fait, les choses sont légèrement plus compliquées ici, car l'impossibilité pour un agent d'en envier un autre est uniquement due à la modélisation du problème, centrée sur les demandes en tant qu'objets indivisibles. Les choses peuvent être différentes si l'on adopte une modélisation fondée sur le partage du temps d'utilisation des satellites.
- ▷ Les préférences des agents peuvent être des ordres complets sur l'ensemble des objets. Considérons par exemple l'application 4 concernant l'allocation de sujets de TREX. Dans ce problème, les agents expriment leurs préférences par classement de l'ensemble des sujets sans *ex-aequo* possible. Puisqu'il y a quasiment autant d'individus que de sujets, un partage sans envie ne

peut exister, sauf s'il est possible de donner à chaque agent le sujet correspondant à son premier choix.

1.3.2 Ordre de bien-être social et fonction d'utilité collective

Notre travail sur les problèmes de partage s'appuie sur les fondements micro-économiques du *welfarisme* et de l'absence d'envie, qui constituent des approches tout-à-fait pertinentes pour traiter les problèmes à portée limitée auxquels nous nous intéressons. Nous allons donc maintenant présenter les notions théoriques qui sont à la base de ces modèles.

Le *welfarisme* idéalise un problème de décision collective en attachant à chaque alternative faisable $x \in \mathscr{E}$ (chaque décision possible, ou encore chaque partage admissible) le vecteur $(u_1, \ldots, u_n) \in \mathscr{V}^n$ des niveaux d'utilité individuelle (par la suite $\mathscr{V}$ sera $\mathbb{N}$, $\mathbb{Q}$ ou encore $\mathbb{R}$, munis de la relation d'ordre habituelle $\geq$), où u_i est l'utilité de l'agent i ($u_i = f_i(x)$, si f_i est la fonction d'utilité de l'agent i). Toute l'information pertinente est donc contenue dans l'ensemble des vecteurs d'utilité faisables. Ces utilités individuelles sont agrégées en une structure de préférence ordinale collective grâce à l'*ordre de bien-être collectif.*

Définition 1.14 (Ordre de bien-être collectif) *Soient $\mathscr{N}$ un ensemble d'agents et $\mathscr{V}$ un espace de valuations. Un* ordre de bien-être collectif *(ou* ordre de bien-être social[7]*) est un préordre $\succeq$ sur $\mathscr{V}^{|\mathscr{N}|}$.*

La notion d'ordre de bien-être collectif est intuitive : chaque agent i possédant une fonction d'utilité f_i de l'espace des alternatives dans $\mathscr{V}$, on peut associer à chaque alternative x un vecteur d'utilités $(u_1, \ldots, u_n) = (f_1(x), \ldots, f_n(x))$. Le rôle de l'ordre de bien-être collectif est de classer les alternatives par le biais de leur vecteur d'utilités associé.

À l'instar de la structure de préférence ordinale, l'ordre de bien-être social possède un équivalent numérique, sous la forme des *fonctions d'utilité collective.*

Définition 1.15 (Fonction d'utilité collective) *Soient $\mathscr{N}$ un ensemble d'agents et $\mathscr{V}$ un espace de valuations. Une* fonction d'utilité collective *est une fonction de $\mathscr{V}^{|\mathscr{N}|}$ dans $\mathscr{V}$.*

Comme pour la représentation numérique de structures d'utilité cardinale, à toute fonction d'utilité collective g est associée un ordre de bien-être social unique $\succeq$ défini comme suit : $\overrightarrow{u} \succeq \overrightarrow{v} \Leftrightarrow g(\overrightarrow{u}) \succeq g(\overrightarrow{v})$. On dit que g représente $\succeq$. De manière évidente, si g représente $\succeq$, pour toute fonction τ croissante, $\tau(g)$ représente aussi $\succeq$.

Nous nous devons de relever l'analogie formelle existant entre le cadre *welfariste* de la décision collective et celui de la décision multicritère, dans lequel la fonction d'utilité collective a un équivalent prenant la forme d'un *opérateur d'agrégation multicritère* [Marichal, 1999] :

Définition 1.16 (Opérateur d'agrégation) *Soient $\mathcal{E}$ et $\mathcal{F}$ deux intervalles de $\mathbb{R}$. Un* opérateur d'agrégation *est une fonction $g_{agg} : \mathcal{E}^n \to \mathcal{F}$.*

Dans notre cas, les critères sont les fonctions d'utilité des agents, et $\mathcal{E}$ et $\mathcal{F}$ sont les espaces (non nécessairement bornés) de valuation. Cette analogie est intéressante d'un point de vue mathématique et informatique, car le domaine de la décision multicritère est historiquement lié au développement de sciences comme la recherche opérationnelle, ou la théorie de la décision. Nous pouvons donc bénéficier des nombreuses avancées dans ces domaines pour l'étude des problèmes de partage. Notons

[7] Par la suite, nous emploierons les deux expressions de manière interchangeable.

toutefois que l'analogie entre décision collective et décision multicritère se limite au point de vue formel, et en particulier toute la réflexion philosophique et éthique à la base de la décision collective est absente de la décision multicritère. On se gardera bien par exemple d'interpréter l'opérateur d'agrégation de critères min en termes autres que celui de l'équilibre entre les critères, et on pourra remarquer aussi que des propriétés telles que l'absence d'envie sont complètement absentes du domaine de la décision multicritère.

Notons, pour clore cette introduction des éléments de base du modèle, l'existence d'un cadre généralisant légèrement celui du *welfarisme* et des ordres de bien-être social : le cadre de la négociation collective (*axiomatic bargaining*, voir par exemple [Moulin, 1988, chapitre 3]). Ce modèle introduit par Nash dans [Nash, 1950], généralise la notion d'ordre de bien-être collectif en introduisant les *fonctions de choix social*. Alors que dans le modèle classique, il est possible de comparer directement deux profils d'utilité sans aucune autre donnée, dans ce nouveau modèle, la comparaison entre deux profils dépend en plus de l'ensemble des alternatives faisables. En fait, l'élément-clef de cette construction est la donnée d'une fonction de choix social qui associe à tout ensemble possible de vecteurs d'utilité admissibles un élément de cet ensemble. Un exemple remarquable de fonction de choix social est la fonction d'*égalitarisme relatif* de Kalai-Smorodinski [Kalai et Smorodinsky, 1975], dont nous reparlerons brièvement lorsque nous aborderons la question de la normalisation des utilités dans la section 1.3.4.5.

1.3.3 Propriétés des ordres de bien-être collectif et des partages optimaux

Le choix de l'ordre de bien-être social ou de la fonction d'utilité détermine le contenu éthique et moral associé à la prise de décision, et implique donc le choix crucial du type de société désiré par les agents. L'introduction d'un ensemble de propriétés associées aux ordres de bien-être collectif et aux décisions optimales impliquées par le choix d'un ordre de bien-être social permet d'aider le décideur à faire son choix parmi les ordres de bien-être collectif classiques.

1.3.3.1 Propriétés basiques

Deux propriétés des ordres collectifs sont généralement requises : l'unanimité et l'anonymat.

Définition 1.17 (Unanimité) *Soit $\succeq$ un ordre de bien-être collectif. $\succeq$ satisfait la propriété d'*unanimité *si et seulement si $\forall (u, v) \in \mathscr{V}^n \times \mathscr{V}^n$, si $\forall i \in \mathscr{N}$, $u_i \geq v_i$ et $\exists j \in \mathscr{N}$ tel que $u_j > v_j$, alors $u \succ v$.*

La propriété d'unanimité est intuitive et souhaitable. Si une alternative est au moins aussi bonne qu'une autre pour l'ensemble des agents et qu'elle est strictement préférée pour au moins un agent, alors elle doit être mieux classée que la première par l'ordre collectif. Le principe d'unanimité est le concept le plus important de la micro-économie. Il ne dépend d'ailleurs pas de la structure de préférences, puisqu'il ne nécessite pas de comparaison interpersonnelle cardinale des préférences. Une autre manière de formuler ce principe est de dire qu'il est compatible avec la relation de dominance de Pareto, c'est-à-dire que u Pareto-domine v si et seulement si $u \succ v$:

Définition 1.18 (Dominance de Pareto, Pareto-efficacité) *Soit $\mathscr{E}$ un ensemble d'alternatives, $\mathscr{N}$ un ensemble d'agents et $(f_1, \ldots, f_n)$ l'ensemble de leurs fonctions d'utilité. Soit $(x, y) \in \mathscr{E}^2$. Si $\forall i \in \mathscr{N}$, $f_i(x) \geq f_i(y)$ et $\exists j \in \mathscr{N}$ tel que $f_j(x) > f_j(y)$, alors x* Pareto-domine *y.*

Une alternative non Pareto-dominée est dite Pareto-efficace.

Dans la plupart des problèmes de décision collective, on souhaite l'égalité des agents devant la procédure de choix, dans le sens où si l'on échange l'identité de deux agents sans changer leurs préférences, le classement résultant des alternatives ne doit pas changer. Cela est garanti par la propriété d'*anonymat* :

Définition 1.19 (Anonymat) *Soit $\succeq$ un ordre de bien-être collectif. $\succeq$ satisfait la propriété d'*anonymat *si et seulement si* $\forall \overrightarrow{u} \in \mathscr{V}^n$, *$\sigma$ permutation de $\mathscr{N}$, et $\overrightarrow{v} \in \mathscr{V}^n$ tel que $\overrightarrow{v} = (u_{\sigma(1)}, \ldots, u_{\sigma(n)})$, on a $\overrightarrow{u} \sim \overrightarrow{v}$.*

Comme pour la propriété d'unanimité, cette propriété d'anonymat est purement ordinale, donc elle est définie quelque soit la structure de préférences employée par les agents. Notons que cette propriété est souvent vue comme la plus fondamentale des propriétés liées à l'équité : elle empêche la discrimination des agents sur des caractéristiques *a priori* hors du cadre du problème de décision en cours.

Nous introduisons enfin une dernière propriété, qui est la clef de la rationalité *welfariste* [Moulin, 2003] : l'indépendance vis-à-vis des agents non concernés (IUA pour *Independance of Unconcerned Agents*) :

Définition 1.20 (Indépendance vis-à-vis des agents non concernés (IUA)) *Soit $\succeq$ un ordre de bien-être collectif. $\succeq$ satisfait la propriété d'*indépendance vis-à-vis des agents non concernés *si et seulement si pour tout quadruplet de profils d'utilité $(\overrightarrow{u}, \overrightarrow{v}, \overrightarrow{u}', \overrightarrow{v}')$ tel que :*
- ▷ *pour un agent i : $u_i = v_i$ et $u'_i = v'_i$,*
- ▷ *pour tout agent $k \neq i$: $u_k = u'_k$, $v_k = v'_k$,*

nous avons : $\overrightarrow{u} \preceq \overrightarrow{v} \Leftrightarrow \overrightarrow{u}' \preceq \overrightarrow{v}'$.

En d'autres termes, tout agent i indifférent vis-à-vis du choix entre deux profils $\overrightarrow{u}$ et $\overrightarrow{v}$ — car son utilité reste la même entre les deux profils — peut être ignoré. Si cette propriété n'est pas respectée, le choix entre deux profils d'utilité va dépendre d'agents qui sont réellement indifférents entre les deux profils, ce qui n'est pas souhaitable. Cette propriété est aussi appelée propriété de «séparabilité». Comme nous le verrons, les seuls ordres de bien-être collectif «continus» respectant ce principe seront les ordres représentés par une fonction d'utilité additive.

1.3.3.2 Partage et équité

Les deux propriétés précédentes sont en général exigées dans n'importe quel type de problème de décision collective. Les définitions que nous allons introduire maintenant sont issues de l'abondante littérature sur l'équité dans le partage et la décision collective. Bien entendu, comme nous l'avons rappelé précédemment, l'équité est un principe flou faisant référence à ce qu'une société juge approprié aux besoins, statuts et contributions de ses membres. Néanmoins, plusieurs critères ont été proposés.

Propriétés relatives au partage Les deux propriétés que nous allons introduire ici sont spécifiques au partage, et ne s'appliquent donc pas de manière générale à des problèmes de décision collective autres issus d'autres domaines que la justice distributive.

La première traduction historique du principe d'équité dans les problèmes de partage est fondée sur l'idée que chaque agent considère «l'utilité qui lui est due» comme étant le n$^{\text{ème}}$ de l'utilité qu'il aurait obtenu s'il était seul. Cette idée apparaît dans [Brams et Taylor, 1996] sous le nom de *proportionnalité*, et dans [Moulin, 1995] sous le nom de *juste part garantie*.

Définition 1.21 (Test de juste part) *Soient $\mathscr{N}$ un ensemble d'agents, $\mathscr{A}$ l'ensemble des partages admissibles et $\overrightarrow{\pi}$ un partage. $\overrightarrow{\pi}$ satisfait le* test de juste part *(fair share en Anglais) si et seulement si $\forall i$, $f_i(\pi_i) \geq \frac{1}{n}\widehat{u_i}$, avec $\widehat{u_i} \overset{def}{=} \max\{f_i(\pi_i) \mid \overrightarrow{\pi} \in \mathscr{A}\}$.*

Il semble assez souhaitable de faire en sorte que l'ordre de bien-être social utilisé fournisse un partage optimale garantissant la juste part aux agents. Cependant, lorsque la ressource à partager est indivisible et qu'aucune compensation monétaire n'est possible entre les agents, il peut n'exister aucun partage admissible satisfaisant le test de juste part [Brams et Taylor, 1996]. Cela se traduit par une propriété souhaitable des ordres de bien-être social, proposée dans [Fargier *et al.*, 2004a; Bouveret *et al.*, 2005b], qui stipule qu'un ordre de bien-être social doit fournir une solution qui satisfait le test de juste part s'il en existe une :

Définition 1.22 (Juste part garantie) *Soient $\mathscr{N}$ un ensemble d'agents, $\mathscr{A}$ l'ensemble des partages admissibles, $\succeq$ un ordre de bien-être collectif et $\widehat{\mathscr{A}}$ l'ensemble des solutions non dominées pour cet ordre collectif. Soit $\mathscr{F} = \{\overrightarrow{\pi} \in \mathscr{A} \mid \forall i \in [\![1, n]\!],\ f_i(\pi_i) \geq \frac{1}{n}\widehat{u_i}\}$. $\succeq$ vérifie la propriété de* juste part garantie *si et seulement si $\mathscr{F} \neq \varnothing \Rightarrow \widehat{\mathscr{A}} \cap \mathscr{F} \neq \varnothing$.*

Cette propriété est donc fondée sur ce à quoi chaque agent estime personnellement avoir droit, sans tenir compte de ce que reçoivent les autres agents.

L'autre vision classique et séduisante de l'équité dans les problèmes de partage est fondée sur la comparaison personnelle (interne à chaque agent) de la propre part d'un agent et de la part des autres agents : il s'agit de l'absence d'envie, que nous avons présentée précédemment. La définition de l'envie, formalisée dans [Foley, 1967], est simple : un agent envie un autre s'il serait plus heureux d'avoir la part de l'autre agent que d'avoir sa propre part.

Définition 1.23 (Test d'absence d'envie) *Soit $\mathscr{N}$ un ensemble d'agents, $\{f_1, \ldots, f_n\}$ l'ensemble de leurs fonctions d'utilités exprimées sur les parts et $\overrightarrow{\pi}$ un partage. $\overrightarrow{\pi}$ satisfait le* test d'absence d'envie *si et seulement si $\forall i \neq j$, $f_i(\pi_i) \geq f_i(\pi_j)$.*

Un partage satisfaisant le test d'absence d'envie est dit sans envie*.*

Bien entendu, comme nous allons le voir à la section 4.1 du chapitre 4, le critère d'absence d'envie seul n'est pas suffisant, car il existe toujours un partage sans envie : le partage qui ne donne rien à personne est sans envie. Les choses se compliquent lorsque l'on ajoute un critère d'*efficacité* à l'absence d'envie : par exemple si l'on requiert que le partage soit complet (attribue l'intégralité de la ressource), ou soit Pareto-efficace.

Dans le cas du partage de biens divisibles, ou dans le cas où les compensations monétaires sont possibles, il existe toujours un partage complet et sans envie, et il existe des procédures pour le trouver dans certains cas (voir [Brams et Taylor, 1996]), par exemple sous réserve de certaines hypothèses sur les fonctions d'utilité des agents, comme pour la procédure de Knaster, ou sur le nombre d'agents en jeu, comme pour la procédure *Adjusted Winner* (dans le cas de deux agents). Quant à l'existence d'un partage Pareto-efficace et sans envie, elle est garantie dans le cas indivisible avec compensation monétaires si les fonctions d'utilité individuelle des agents ont une certaine forme (par exemple si elles sont superadditives [Alkan *et al.*, 1991]). Dans le cas du partage de biens indivisibles sans compensation monétaire, il n'existe pas toujours de partage efficace et sans envie, et comme nous allons le montrer au chapitre 4, et la seule tâche de démontrer l'existence d'un partage sans envie et efficace peut s'avérer extrêmement complexe.

Exemple 1.1 Soit le partage à 2 objets o_1 et o_2 et 2 agents 1 et 2. Les préférences des agents sont les suivantes : 1 et 2 ont les mêmes préférences et valuent $\varnothing$ à 0, $\{o_1\}$ à 5, $\{o_2\}$ à 6 et $\{o_1, o_2\}$ à 10. Alors :

▷ Le partage $(\varnothing, \{o_1, o_2\})$ n'est pas sans-envie (1 envie 2),
▷ Le partage $(\{o_1\}, \{o_2\})$ n'est pas sans-envie (1 envie 2),
▷ Le partage $(\{o_2\}, \{o_1\})$ n'est pas sans-envie (2 envie 1),
▷ Le partage $(\{o_1, o_2\}, \varnothing)$ n'est pas sans-envie (2 envie 1).

Dans cet exemple, il n'existe aucun partage complet sans-envie. En revanche, si les compensations monétaires sont possibles, dans tous les partages, l'agent bénéficiaire de la plus grande part peut reverser la moitié de son utilité sous forme d'argent à l'agent lésé, produisant ainsi des partages sans envie.

Équité fondée sur l'égalitarisme et mesures d'inégalité Les traductions de l'équité introduites jusqu'ici sont d'une part spécifiques au partage, et d'autre part ne requièrent par de comparaison interpersonnelle des utilités, ce qui leur confère un intérêt particulier. En revanche, ces critères ne tirent pas réellement partie des hypothèses très fortes à la base du *welfarisme* cardinal, et liées au fait que les préférences des agents sont exprimées sur une échelle numérique.

Une traduction très largement acceptée de la notion d'équité dans la micro-économie est fondée sur l'égalitarisme. La notion d'équité est traduite par l'aspiration à tendre vers une égalité parfaite des utilités individuelles, si tant est que les agents ont des droits égaux sur la ressource et que les utilités sont commensurables, c'est-à-dire exprimées sur des échelles identiques (monétaires par exemple). Dans ce cadre, on peut déterminer de manière précise à quel point un partage est inéquitable, en mesurant la «distance» du profil d'utilité en question au profil d'utilité parfaitement égalitaire. Cette traduction de la notion d'équité a donné naissance, notamment sous l'impulsion de [Atkinson, 1970], à une branche très prolifique de la micro-économie : celle de la mesure des inégalités. De manière intéressante, le domaine de la mesure des inégalités n'est pas exclusivement réservé à la décision collective, mais il est aussi abondamment étudié dans le contexte de la décision multicritère [Keeney et Raiffa, 1976], car de nombreux problèmes nécessitent la recherche d'un certain équilibre entre différents critères, et les mêmes outils formels s'appliquent dans ce cas-là. En outre, la notion d'inégalité est aussi formellement très proche de la notion de risque dans le domaine de la prise de décision en présence de risque. Nous aurons l'occasion de revenir sur ce point particulier dans le chapitre 2.

L'équité égalitariste et la mesure des inégalités sont fondées sur la propriété de réduction des inégalités. Cette propriété caractérise l'incitation à l'équité d'un ordre de bien-être collectif par sa tendance à redistribuer les utilités des agents les plus riches vers les agents les plus pauvres. Cette notion s'appuie sur la définition d'un *transfert de Pigou-Dalton* :

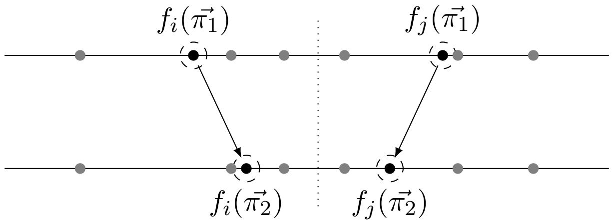
Définition 1.24 (Transfert de Pigou-Dalton (réduction des inégalités)) *Soient $\overrightarrow{u}$ et $\overrightarrow{u}'$ deux profils d'utilité. $\overrightarrow{u}'$ est obtenu à partir de $\overrightarrow{u}$ par* réduction des inégalités*, ou* transfert de Pigou-Dalton *si et seulement si $\exists (i,j) \in \mathscr{N}^2$ tels que :*
▷ $i \neq j$;
▷ $\overrightarrow{u_i} + \overrightarrow{u_j} = \overrightarrow{u_i}' + \overrightarrow{u_j}'$ *(conservation de la somme)* ;
▷ $u_i < \{u_i', u_j'\} < u_j$ *(réduction des inégalités)* ;
▷ $\forall k \in \mathscr{N} \setminus \{i, j\}$, $u_k = u_k'$.

En d'autres termes, un transfert de Pigou-Dalton redistribue l'utilité d'un agent riche j vers un agent i (tout en maintenant l'utilité globale constante), sans modifier l'utilité des autres agents. À partir de cette notion, on peut caractériser la tendance qu'a un ordre de bien-être social à favoriser l'équité, en introduisant la propriété suivante :

Définition 1.25 (Principe de réduction des inégalités) *Soit $\succeq$ un ordre de bien-être collectif. $\succeq$ satisfait le* principe de réduction des inégalités *si et seulement si pour toute paire de profils d'utilité*

$\overrightarrow{u}$, $\overrightarrow{u}'$ *tels que* $\overrightarrow{u}'$ *est obtenu à partir de* $\overrightarrow{u}$ *par transfert de Pigou-Dalton, on a* $\overrightarrow{u} \prec \overrightarrow{u}'$.

Ce principe est aussi appelé *principe de Pigou-Dalton* dans la littérature (voir par exemple [Moulin, 1988, page 45] ou [d'Aspremont et Gevers, 2002, page 506]). Si la préférence pour le second partage est large, l'ordre collectif respecte *faiblement* le principe de réduction des inégalités. La notion de transfert de Pigou-Dalton est illustrée sur la figure 1.4.



▲ Réduction des inégalités entre deux profils d'utilité.

Situation des profils moins inégalitaires pour deux agents. ►

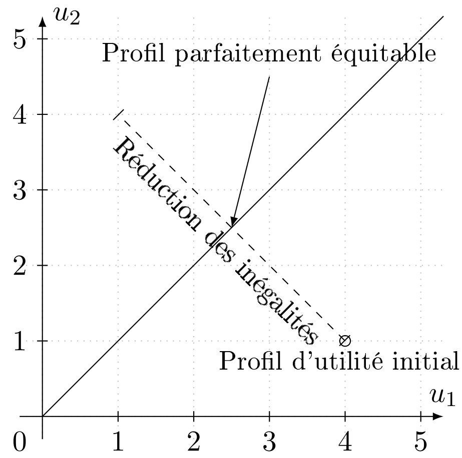


Figure 1.4 — Illustration du principe de réduction des inégalités de Pigou-Dalton.

La notion de réduction des inégalités seule est en général insuffisante pour caractériser l'ensemble des décisions collectives «intéressantes», car elle n'implique pas en particulier que la décision collective soit Pareto-optimale. Cependant, il existe un outil très intéressant, la courbe de Lorenz, qui fournit une relation de dominance entre profils d'utilité permettant de prendre en compte à la fois la notion de Pareto-efficacité et la réduction des inégalités :

Définition 1.26 (Courbe de Lorenz) *Soit* $\overrightarrow{u}$ *un vecteur d'utilités et* $\overrightarrow{u}^{\uparrow}$ *le vecteur des composantes ordonnées par ordre non décroissant de* $\overrightarrow{u}$ *(on notera* $u_k^{\uparrow}$ *la* $k^{ème}$ *composante de ce vecteur). Alors la* courbe de Lorenz *de* $\overrightarrow{u}$ *est le vecteur* $\overrightarrow{L(\overrightarrow{u})} = (u_1^{\uparrow}, \ldots, \sum_{k=1}^{i} u_k^{\uparrow}, \ldots, \sum_{k=1}^{n} u_k^{\uparrow})$.

La courbe de Lorenz d'un profil d'utilité est donc définie comme le vecteur qui à tout indice i comporte la somme de toutes les utilités des i agents les moins satisfaits. Cet outil, transposé à l'échelle d'une population et appliqué au vecteur des revenus des individus, est utilisé en économie pour mesurer le taux d'inégalité au sein d'une population. Par exemple, la composante de la courbe de Lorenz correspondant à 30% de la taille de la population représente la somme des revenus des 30% des individus qui sont les plus pauvres de la population. Cet outil a l'avantage d'être illustratif du point de vue graphique : la représentation graphique d'une courbe de Lorenz est toujours «convexe» (voir figure 1.6), et son degré de convexité indique l'inégalité au sein de la population. Si tous les agents sont complètement égaux, la courbe est linéaire, et à l'extrême si tous les agents sauf un ont une utilité nulle, la courbe est la plus éloignée possible d'une droite. Notons que le vecteur correspondant à notre définition de la courbe de Lorenz est parfois appelée *courbe de Lorenz généralisée*, la courbe de Lorenz étant définie dans ce contexte comme la normalisation de la courbe de Lorenz généralisée (c'est-à-dire correspondant au vecteur introduit dans la définition 1.26 dans lequel chaque composante a été divisée par la somme totale des utilités).

Comme nous l'avons fait remarquer, la notion de courbe de Lorenz concentre dans un seul critère la Pareto-efficacité et la réduction des inégalités :

Proposition 1.2 ([Moulin, 1988]) *On dira qu'un vecteur d'utilités* $\overrightarrow{u}$ Lorenz-domine *un vecteur d'utilités* $\overrightarrow{v}$ *si sa courbe de Lorenz* $\overrightarrow{L(\overrightarrow{u})}$ *Pareto-domine* $\overrightarrow{L(\overrightarrow{v})}$.

Si $\overrightarrow{u}$ *Pareto-domine* $\overrightarrow{v}$ *ou bien est obtenu par transfert de Pigou-Dalton à partir de* $\overrightarrow{v}$*, alors* $\overrightarrow{u}$ *Lorenz-domine* $\overrightarrow{v}$*. Réciproquement, si* $\overrightarrow{u}$ *Lorenz-domine* $\overrightarrow{v}$*, alors il existe une suite de transferts de Pigou-Dalton et d'améliorations de Pareto qui transforme* $\overrightarrow{v}$ *en* $\overrightarrow{u}$.

En conséquence, un ordre collectif respecte le principe de réduction des inégalités s'il est compatible avec la dominance de Lorenz. L'ensemble des partages non dominés au sens de Lorenz (Lorenz-optimaux) est le sous-ensemble «le plus égalitaire au sens de Pigou-Dalton» de l'ensemble des partages Pareto-optimaux. La figure 1.5 illustre la notion de dominance de Lorenz sur un ensemble de profils d'utilité pour un problème à deux agents.

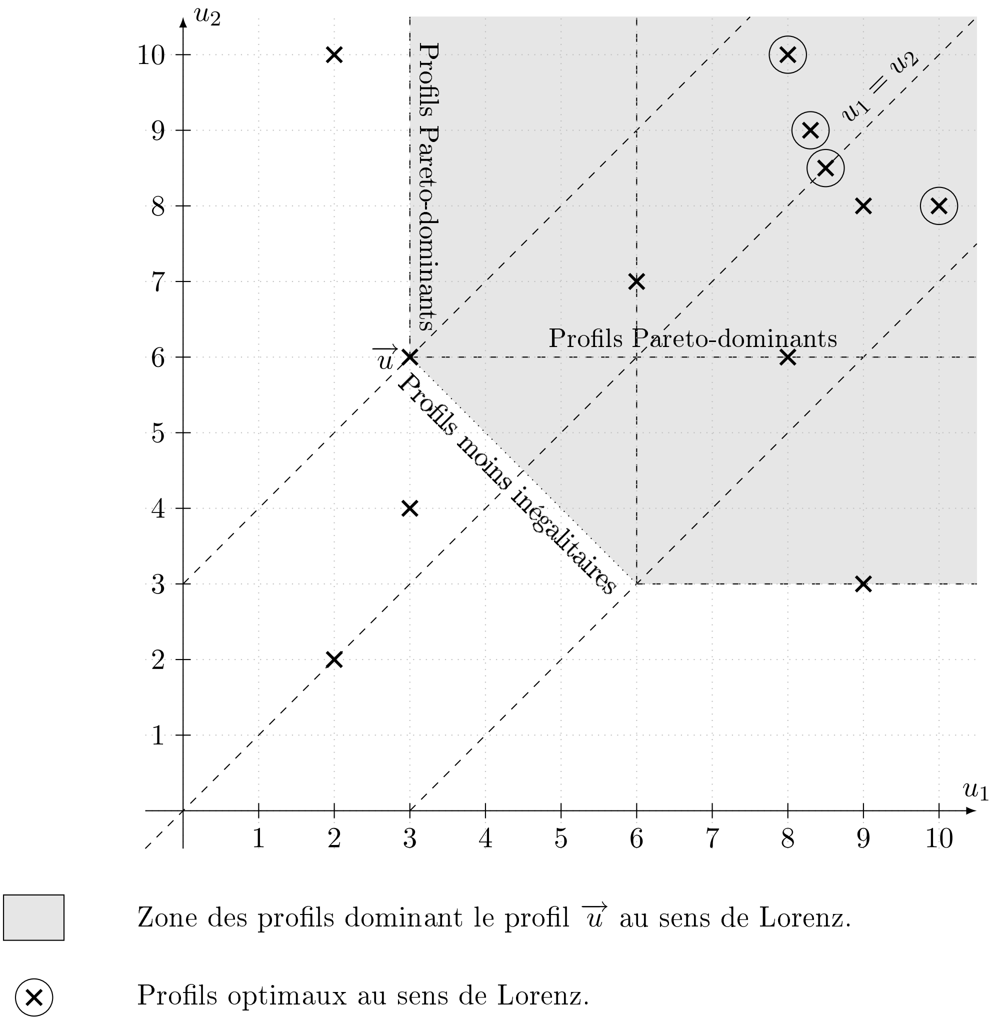


Figure 1.5 — Illustration de la notion de dominance de Lorenz sur des profils d'utilité à deux composantes.

Indices d'inégalité Si les outils fournis par le test de réduction des inégalité et la courbe de Lorenz formalisent la notion d'inégalité, ils n'indiquent pas comment celle-ci doit être mesurée concrètement. Plusieurs mesures numériques ont été proposées, sous la forme d'*indices d'inégalité*. Un indice d'inégalité est une transformation mathématique d'une fonction d'utilité collective qui met en valeur la perte de bien-être social due à l'inégalité entre agents. [Moulin, 1988] propose une bonne introduction aux indices d'inégalité.

Définition 1.27 (Indice d'inégalité) *Soit $\succeq$ un ordre de bien-être collectif qui respecte le principe de réduction des inégalités. Pour chaque vecteur d'utilité positif $\overrightarrow{u}$ on définit l'utilité également distribuée équivalente $\varepsilon(\overrightarrow{u}) \in \mathscr{V}$ de la manière suivante : $\varepsilon(\overrightarrow{u}) \cdot (1, \ldots, 1) \sim \overrightarrow{u}$. On note également $\overline{u} = \frac{1}{n}\sum_{i=1}^{n} u_i$. L'*indice d'inégalité *associé à $\succeq$ est :*

$$J(\overrightarrow{u}) = 1 - \frac{\varepsilon(\overrightarrow{u})}{\overline{u}}.$$

On peut remarquer que $J(\overrightarrow{u})$ est toujours strictement positif, sauf lorsque toutes les composantes de $\overrightarrow{u}$ sont égales, auquel cas il est nul. La positivité est due au fait que $\succeq$ respecte le principe de réduction des inégalités. En outre, $J(\overrightarrow{u}) \leq 1$.

Concrètement, un indice d'inégalité est donc fondé sur la mesure d'une «distance» entre un profil d'utilité et le profil «parfait» (c'est-à-dire parfaitement égalitaire) qui lui est équivalent selon l'ordre de bien-être social choisi pour la construction de l'indice. Selon la définition de la fonction d'utilité collective ou de l'ordre collectif choisis au départ, on aboutit à des indices d'inégalité très différents. On peut citer les deux exemples les plus classiques : les indices d'Atkinson et l'indice de Gini.

La famille d'indices d'Atkinson est fondée sur la famille de fonctions d'utilité collective somme des puissances que nous introduirons dans la section 1.3.4, restreinte aux fonctions qui respectent le principe de réduction des inégalités.

Définition 1.28 (Indices d'Atkinson) *La famille d'indices d'Atkinson est la famille définie par :*

$$\begin{cases} J_q(\overrightarrow{u}) = 1 - \left(\dfrac{1}{n}\displaystyle\sum_{i=1}^{n}\left(\frac{u_i}{\overline{u}}\right)^q\right)^{\frac{1}{q}}, & 0 < q < 1 \text{ ou } q < 0 \\ J_0(\overrightarrow{u}) = 1 - \left(\displaystyle\prod_{i=1}^{n}\frac{u_i}{\overline{u}}\right)^{\frac{1}{n}}. \end{cases}$$

L'indice de Gini, quant à lui, n'est pas fondé sur une fonction d'utilité collective classique, mais sur la mesure de la distance de la courbe de Lorenz d'un vecteur d'utilités à sa courbe idéale (c'est-à-dire la droite $k \mapsto k\overline{u}$). Plus le vecteur d'utilités est inégalitaire, plus cette «distance» sera grande. L'indice de Gini mesure l'aire de la surface comprise entre la courbe de Lorenz réelle et idéale (aire grisée sur la figure 1.6).

Définition 1.29 (Indice de Gini) *L'indice d'inégalité de Gini se définit par les trois expressions équivalentes suivantes :*

$$G(\overrightarrow{u}) = \frac{\displaystyle\sum_{k=1}^{n}(k\overline{u} - L(\overrightarrow{u})_k)}{\displaystyle\frac{n}{2}\sum_{i=1}^{n}u_i} = 1 - \frac{1}{n^2\overline{u}}\left(\sum_{k=1}^{n}(2(n-k)+1)u_k^{\uparrow}\right) = \frac{1}{2n^2\overline{u}}\sum_{1\leq i,j\leq n}|u_i - u_j|.$$

Les trois définitions équivalentes de l'indice de Gini suggèrent trois interprétations.

1. La première définition correspond au calcul normalisé de l'aire décrite ci-avant, et grisée sur la figure 1.6.

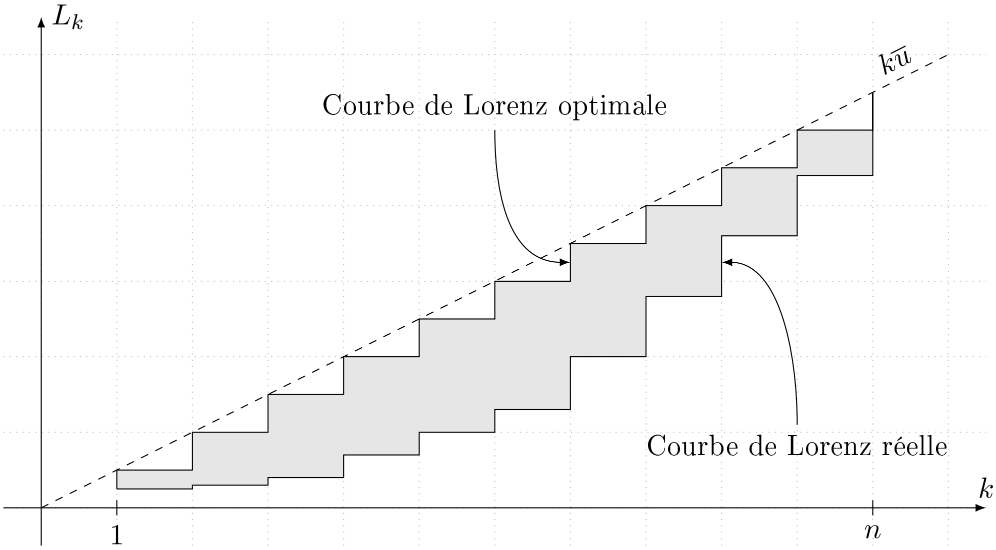


Figure 1.6 — Indice de Gini : distance entre les courbes de Lorenz réelle et idéale.

2. La deuxième définition fait apparaître la fonction d'utilité collective sur laquelle est construite l'indice de Gini. Cette fonction est une variante de la fonction d'utilité utilitariste (voir section 1.3.4) dans laquelle le poids d'un agent décroît en fonction de son degré de satisfaction par rapport aux autres (en fait, c'est une moyenne pondérée ordonnée, voir même section).
3. La troisième interprétation est fondée sur les utilités différentielles : l'indice de Gini est la moyenne des différences d'utilités deux-à-deux entre agents.

1.3.3.3 Résumé de l'ensemble des propriétés

Nous avons introduit un certain nombre de propriétés permettant de caractériser les partages (ou décisions collectives), et les ordres de bien-être social. La plupart de ces propriétés sont liées à la définition de l'équité, selon plusieurs points de vue différents. Un récapitulatif de l'ensemble de ces propriétés est proposé dans le tableau 1.1.

Propriété de l'ordre de bien-être collectif	Propriété du partage	Spécifique au partage ?	comparaison interpersonnelle des préférences ?	Critère d'équité ?
Unanimité	Pareto-efficacité	non	non	non
Anonymat	(—)	non	non	oui/non*
IUA	(—)	non	non	oui/non*
Juste part garantie	Test de juste part	oui	non	oui
(—)	Absence d'envie	oui	non	oui
Réduction des inégalités	Lorenz-efficacité	non	oui	oui
Indice d'inégalité	(—)	non	oui	oui

* Selon les points de vue, il peut s'agir ou non d'un critère d'équité.

Tableau 1.1 — Récapitulatif des propriétés des ordres collectifs et des partages

1.3.4 Fonctions d'utilité collective classiques

Nous allons introduire dans cette section l'ensemble des fonctions d'utilité collective les plus classiques, que nous analyserons brièvement à la lumière des propriétés précédentes.

1.3.4.1 Fonction d'utilité collective utilitariste classique

Les deux fonctions d'utilité collective à la base de toute l'analyse micro-économique correspondent aux deux visions contradictoires présentées ci-avant : la théorie de l'utilitarisme classique et celle de l'égalitarisme. La première de ces deux visions a conduit à définir de manière naturelle la fonction somme comme fonction d'agrégation des utilités individuelles. La fonction d'utilité collective utilitariste classique est issue d'une certaine idée de la justice collective : la justice selon l'adéquation. Chaque agent produisant une part d'utilité collective, si un agent est plus productif qu'un autre, et seulement dans ce cas, il aura le droit à plus de ressource.

Définition 1.30 (Utilité collective utilitariste classique) *La* fonction d'utilité collective utilitariste classique *est la fonction de $\mathscr{V}^n$ dans $\mathscr{V}$: $g^\star : (u_1, \ldots, u_n) \mapsto \sum_{i=1}^n u_i$.*

Cette fonction d'utilité collective est intéressante du point de vue de l'efficacité : il est apparent qu'elle satisfait le principe d'unanimité. En outre, elle garantit aussi l'anonymat, l'indépendance vis-à-vis des agents non concernés, et une propriété d'*insensibilité à une dilatation linéaire commune des utilités*. Cette dernière propriété, appelée *independance of common utility scale* dans la littérature anglophone, exprime simplement le fait qu'une transformation linéaire de tous les profils d'utilité ne change pas leur ordre, ou en d'autres termes : pour tout $\lambda > 0$, $\overrightarrow{u} \preceq \overrightarrow{v} \Leftrightarrow \lambda\overrightarrow{u} \preceq \lambda\overrightarrow{v}$. Cette propriété est partagée par toutes les fonctions de la famille somme des puissances (introduite plus loin dans cette section).

Si la fonction utilitariste classique possède quelques bonnes propriétés, elle est en revanche assez peu intéressante du point de vue de l'équité, si toutefois par «équité» on entend égalité entre les agents. Elle ne garantit pas la juste part, et ne réduit pas les inégalités (voir [Moulin, 1988]) — notons qu'elle ne les augmente pas non plus, elle y est indifférente. Dans tous les cas, utiliser une telle fonction d'agrégation peut conduire à des partages très inégalitaires : dans un cas extrême, on peut avoir à choisir entre les profils $(100, 0)$ et $(49, 50)$. La fonction d'utilité utilitariste classique choisira le premier profil, de loin le plus inégalitaire des deux.

Fonction d'utilité collective égalitariste La deuxième fonction d'utilité collective la plus classique correspond à la vision égalitariste de la justice collective. Contrairement à la fonction utilitariste, cette fonction attribue les biens selon les besoins, et non selon la productivité. Elle tend à égaliser le vecteur des utilités individuelles, et n'hésite pas à sacrifier la satisfaction d'un grand nombre d'agents au profit du moins riche.

Définition 1.31 (Utilité collective égalitariste) *La* fonction d'utilité collective égalitariste *est la fonction de $\mathscr{V}^n$ dans $\mathscr{V}$: $g^{(e)} : (u_1, \ldots, u_n) \mapsto \min_{i=1}^n u_i$.*

La fonction d'utilité égalitariste a une particularité intéressante. Elle satisfait la propriété d'*insensibilité à une dilatation commune croissante quelconque des utilités* (*independance of common utility pace*), c'est-à-dire que l'on a $g^{(e)}(u_1, \ldots, u_n) \leq g^{(e)}(v_1, \ldots, v_n) \Leftrightarrow g^{(e)}(\tau(u_1), \ldots, \tau(u_n)) \leq g^{(e)}(\tau(v_1), \ldots, \tau(v_n))$ pour toute transformation τ croissante non nécessairement linéaire. Les utilités individuelles peuvent donc subir n'importe quelle transformation croissante sans changer l'ordre des profils d'utilité.

La fonction d'utilité égalitariste satisfait de même certaines propriétés d'équité telles que l'anonymat ou la juste part garantie, mais en revanche, à l'instar de la fonction utilitariste, elle est indifférente aux inégalités. Cette fonction a un autre problème autrement plus important : elle ne satisfait pas le principe le plus basique d'unanimité — en fait, elle le satisfait au sens *faible* (c'est-à-dire que dans la définition de l'unanimité, $u \succeq v$ au lieu que $u \succ v$). En outre, elle ne satisfait pas non plus l'indépendance vis-à-vis agents non concernés. Ces effets néfastes du min (indifférence aux inégalités, non satisfaction du principe d'unanimité et dépendance vis-à-vis des agents non concernés) sont quelquefois appelés «effet de noyade» voir [Fargier *et al.*, 1993; Dubois et Fortemps, 1999] et sont dus à l'idempotence de l'opérateur min qui se concentre donc sur une seule composante et néglige la comparaison des autres. Considérons par exemple les vecteurs $(1, 1000, \ldots, 1000)$ et $(1, \ldots, 1)$: la fonction d'utilité collective égalitariste laisse ces deux vecteurs indifférents, alors que le premier est clairement meilleur que le second. Ces lacunes n'en font pas une fonction d'utilité collective très pertinente en l'état.

Il existe un raffinement classique connu de cette fonction, qui pallie ces lacunes : l'ordre collectif *leximin*. Cet ordre a été introduit dans [Sen, 1970], en relation avec les travaux de [Rawls, 1971] Il a été repris de nombreuses fois, notamment par [Kolm, 1972; d'Aspremont et Gevers, 1977; Moulin, 1988].

Définition 1.32 (préordre leximin [Sen, 1970; Kolm, 1972]) *Soient $\overrightarrow{u}$ et $\overrightarrow{v}$ deux vecteurs d'utilité de $\mathscr{V}^n$. Ils sont indifférents pour le préordre* leximin *si et seulement si $\overrightarrow{u}^{\uparrow} = \overrightarrow{v}^{\uparrow}$. $\overrightarrow{u}$ est préféré strictement à $\overrightarrow{v}$ (noté $\overrightarrow{u} \succ_{\text{leximin}} \overrightarrow{v}$) si et seulement si $\exists k \in [\![0, n-1]\!]$ tel que $\forall i \in [\![1, k]\!]$, $u_i^{\uparrow} = v_i^{\uparrow}$ et $u_{k+1}^{\uparrow} > v_{k+1}^{\uparrow}$.*

Le préordre leximin, s'il est classique dans le domaine de la décision collective, l'est aussi dans le domaine de la logique floue ([Dubois et Fortemps, 1999]). Dans ce dernier domaine, un autre raffinement classique du min est souvent introduit : l'ordre discrimin. Là où l'ordre leximin compare deux alternatives grâce à la comparaison de leurs rangs d'utilité par ordre croissant, l'ordre discrimin utilise la relation d'inclusion : une alternative est préférée à une autre si à un rang donné l'ensemble des agents ayant cette valeur d'utilité de la première alternative est strictement inclus dans ce même ensemble pour l'autre alternative (et si pour les rangs inférieurs ces ensembles sont égaux). Ce raffinement paraît moins pertinent dans le contexte du partage équitable : d'une part, le préordre induit n'est pas total et laisse de nombreuses alternatives incomparables, et d'autre part il ne satisfait pas le principe d'anonymat, en laissant des profils permutés incomparables entre eux au lieu de les laisser indifférents.

L'ordre collectif leximin fonctionne en comparant les utilités des agents les plus pauvres des deux partages. Si ces utilités sont égales, on compare les utilités des prochains dans l'ordre d'utilité croissante, jusqu'à trouver une différence. Cet ordre est le raffinement efficace de la fonction d'utilité collective égalitariste, dans le sens où l'ensemble des solutions admissibles non dominées pour l'ordre leximin est inclus dans l'ensemble des solutions maximisant la fonction d'utilité égalitariste. En conséquence, le leximin possède toutes les bonnes propriétés héritées de la fonction min : anonymat, insensibilité à une dilatation commune croissante quelconque des utilités, juste part garantie. Elle vérifie en plus la propriété de réduction des inégalités, l'indépendance vis-à-vis des agents non concernés, et enfin l'unanimité, ce qui en fait un ordre tout-à-fait pertinent pour l'agrégation d'utilités en une décision collective.

Le préordre leximin possède de plus une propriété remarquable, ce qui explique le fait qu'il occupe une place aussi centrale dans la théorie du *welfarisme* cardinal. C'est en effet le seul ordre de bien-être collectif qui respecte à la fois le principe de réduction des inégalités et l'insensibilité à une dilatation commune croissante quelconque des utilités (voir par exemple [Moulin, 1988, page 40] ou [Moulin, 2003, page 76]).

Nous pouvons cependant remarquer que nous avons défini le critère leximin comme un ordre de bien-être social, et non comme une fonction d'utilité collective, ce qui conduit à l'interrogation légitime : est-il possible de représenter l'ordre de bien-être social leximin par une fonction d'utilité collective ? La réponse est connue (voir par exemple [Moulin, 1988]) et négative dans le cas général :

Proposition 1.3 (voir [Moulin, 1988]) *L'ordre leximin n'est pas représentable par une fonction d'utilité collective, à moins que l'ensemble des utilités ne soit fini ou infini dénombrable.*

Ce résultat négatif n'est pas vraiment limitatif dans le cas général, car dans tous les problèmes concrets que nous aurons à traiter, l'ensemble des alternatives (donc des profils d'utilité) sera bien entendu fini. Dans ce cas précis, il existe des fonctions d'utilité collective permettant de représenter l'ordre leximin, comme nous allons le voir dans le chapitre 5 consacré à l'algorithmique du leximin. Nous aurons cependant à nous poser la question de la pertinence et de l'efficacité opérationnelle liée à la traduction de l'ordre leximin en fonction d'utilité collective.

1.3.4.2 Fonction de Nash

Entre les fonctions d'utilité plutôt extrêmes que sont les fonctions utilitaristes et égalitaristes, on peut définir un certain nombre de fonctions intermédiaires permettant de réaliser des compromis entre ces points de vue. D'un point de vue «philosophique», ces fonctions intermédiaires permettent de réaliser un compromis entre le principe de compensation (égalitarisme) et le principe d'adéquation (utilitarisme).

La première de ces fonctions, moins utilisée que les deux précédentes dans la littérature, mais possédant de bonnes propriétés, est la fonction de Nash :

Définition 1.33 (Utilité collective de Nash) *La* fonction d'utilité collective de Nash *est la fonction de* $\mathscr{V}^n$ *dans* $\mathscr{V}$: $g^{(N)} : (u_1, \ldots, u_n) \mapsto \prod_{i=1}^n u_i$.

Cette fonction a l'avantage de présenter la propriété d'être *indépendante vis-à-vis des échelles individuelles d'utilité*, ce qui signifie que l'échelle sur laquelle est exprimée la satisfaction d'un agent ne compte pas dans le choix de la décision collective, contrairement à l'utilitarisme qui accorde de l'importance seulement aux agents les plus producteurs d'utilité, et à l'égalitarisme qui accorde de l'importance aux plus pauvres uniquement. La fonction de Nash apparaît comme un compromis séduisant dans certains problèmes tels que les problèmes du type partage de temps d'utilisation d'une ressource avec externalités (voir l'exemple de la radio dans [Moulin, 1988, page 80]). En outre, elle est compatible avec l'ordre de Pareto (elle vérifie l'unanimité), l'indépendance vis-à-vis des agents non concernés, et elle réduit les inégalités. Nous aurons l'occasion de revenir sur cette fonction d'utilité dans le chapitre 2.

1.3.4.3 Somme des puissances

La véritable puissance du cadre du *welfarisme* cardinal réside en partie dans la possibilité d'exprimer des compromis paramétrables entre les fonctions égalitariste et utilitariste classique, par le biais de familles de fonctions. Parmi les familles permettant de représenter un large spectre d'ordres de bien-être social, l'une d'elle a été introduite dans un théorème de [Roberts, 1980].

Définition 1.34 (Famille somme des puissances) *La famille de fonctions d'utilité collective* somme des puissances *est la famille :*

$$\forall p \in \mathbb{R}\ g^{(p)}(\overrightarrow{u}) = \begin{cases} sgn(p) \cdot \sum_{i=1}^{n} u_i^p & pour\ p \neq 0 \\ \sum_{i=1}^{n} \log u_i & pour\ p = 0 \end{cases},\ avec\ \forall p \neq 0,\ sgn(p) = \frac{p}{|p|}.$$

Cette famille de fonctions est intéressante à plusieurs points de vue. Tout d'abord, remarquons qu'elle est «continue» en 0, en ce qui concerne les ordres de bien-être collectif représentés. En effet, $\forall a > 0,\ a^x = 1 + x\log(a) + o_{x\to 0}(x^2)$, donc $\sum_{i=1}^n u_i^p = \sum_{i=1}^n (1 + p\log(u_i)) + o_{p\to 0^+}(p^2) = n + p\sum_{i=1}^n \log(u_i) + o_{p\to 0^+}(p^2)$. La fonction $x \mapsto n+px$ étant croissante pour $p > 0$, $n+p\sum_{i=1}^n \log(u_i)$ représente le même ordre de bien-être social que $\sum_{i=1}^n \log u_i$. Le même raisonnement montre la continuité des ordres de bien-être social lorsque $p \to 0^-$.

Ces fonctions ont une autre propriété intéressante. Le théorème de [Roberts, 1980] montre que tous les ordres de bien-être collectifs vérifiant l'anonymat, continus, et séparables (vérifiant la propriété d'indépendance vis-à-vis des agents non concernés) peuvent être représentés par une fonction d'utilité collective de cette famille. Il faut préciser toutefois ce que l'on entend par «continuité» d'un ordre de bien-être collectif : un ordre de bien-être collectif $\succeq$ est dit continu si et seulement si pour tout profil $\overrightarrow{u}$ les ensembles $\{\overrightarrow{v} \mid \overrightarrow{v} \preceq \overrightarrow{u}\}$ et $\{\overrightarrow{v} \mid \overrightarrow{u} \preceq \overrightarrow{v}\}$ sont fermés pour une topologie de l'ensemble $\mathscr{V}$ (en général on suppose que $\mathscr{V} = \mathbb{R}$ et bien entendu la topologie choisie est celle de l'ensemble des ouverts sur les réels). En outre, comme nous l'avons précisé précédemment, toutes les fonctions de cette famille vérifient la propriété d'insensibilité à une dilatation linéaire commune des utilités.

Cette famille somme des puissances fait la jonction entre des fonctions très inégalitaires (lorsque $p > 1$, $g^{(p)}$) augmente les inégalités, et des fonctions plus équitables : lorsque $p \to -\infty$, l'ordre de bien-être social représenté tend vers l'ordre leximin. Nous avons en conséquence la proposition intéressante suivante :

Proposition 1.4 *Si l'ensemble des profils d'utilité est fini, alors il existe un $p < 0$ tel que $g^{(p')}$ représente l'ordre* leximin *pour tout $p' \leq p$.*

La preuve de cette proposition est détaillée à la section B.2.2 de l'annexe B. Il s'avère en revanche plus ardu de trouver l'exposant p maximum tel que $g^{(p')}$ représente l'ordre leximin pour tout $p' \leq p$. Un début d'étude de ce problème est présenté en section B.3 de la même annexe.

À la jonction entre les fonctions réduisant les inégalités ($p < 1$) et les fonctions les augmentant ($p > 1$), se trouve la fonction d'utilité collective utilitariste $g^{(1)}$. De même, à la jonction entre les fonctions qui avantagent les producteurs d'utilité — que l'on pourrait appeler *pseudo-utilitaristes* ($p > 0$) — et les fonctions qui avantagent les agents les plus pauvres — selon la même terminologie, *pseudo-égalitaristes* ($p < 0$) — se trouve l'ordre de bien-être social représenté par la fonction de Nash, $g^{(0)}$, qui est insensible aux échelles des utilités individuelles. Il est à noter que dans la littérature, on restreint souvent la famille somme des puissances aux indices $p \leq 1$, négligeant de fait explicitement les fonctions d'utilité qui augmentent strictement les inégalités.

La figure 1.7 montre la représentation graphique de quelques fonctions puissances, permettant d'illustrer de manière intuitive le principe d'avantage aux riches ou aux pauvres selon la convexité ou la concavité de la courbe. Plus l'exposant p tend vers $-\infty$, plus la fonction $g^{(p)}$ accorde de l'importance aux incréments d'utilité d'un agent pauvre (dont l'utilité est proche de 0), par rapport aux incréments d'utilité d'un agent riche. Cela se traduit graphiquement par la concavité très accentuée de la courbe $u \mapsto u^p$. À l'inverse, si $p > 1$, la convexité de la courbe $u \mapsto u^p$ illustre le fait que plus un agent est riche, plus il sera incité à devenir riche, car plus son utilité augmente, plus un incrément unitaire de son utilité individuelle a d'effet positif sur l'utilité.

La figure 1.8 montre un exemple de courbes iso-utilité collective de quatre fonctions de la famille somme des puissances n'augmentant pas les inégalités[8] pour un problème à deux agents, en fonction de l'utilité individuelle de ces deux agents. On peut remarquer que plus p tend vers $-\infty$, plus les courbes iso-utilité se rapprochent d'une courbe «iso-min».

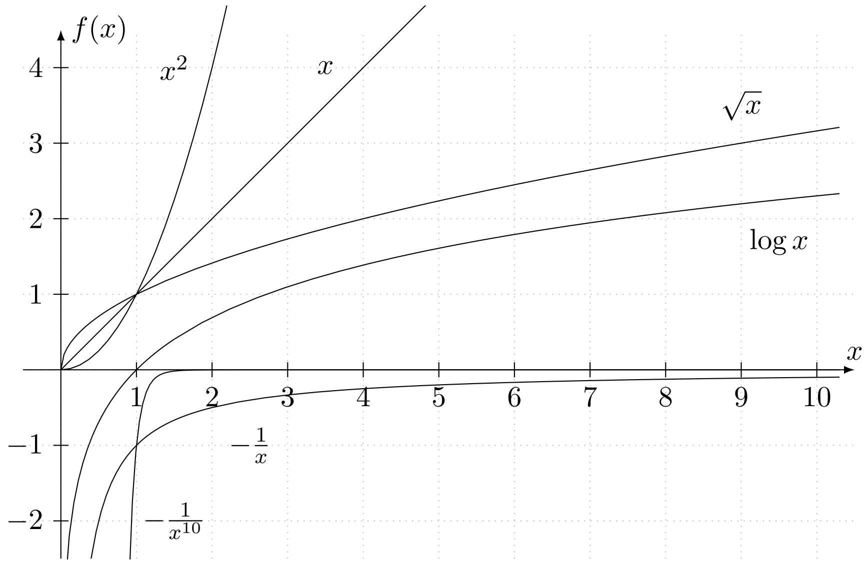


Figure 1.7 — La représentation graphique de quelques fonctions puissance.

1.3.4.4 Moyennes pondérées ordonnées (OWA)

La famille de fonctions somme des puissances était fondée sur des propriétés analytiques fortes : la continuité et la séparabilité, qui équivalent à peu de choses près à une notion de dérivabilité des fonctions. Une seconde famille est très utilisée dans le domaine de l'agrégation de fonctions d'utilité : celle des moyennes pondérées ordonnées [Yager, 1988], ou OWA (pour *Ordered Weighted Averages*). L'idée à la base de la construction de ces fonctions d'utilité est d'introduire une famille permettant de pondérer l'importance des agents non pas selon leur identité, mais selon la place de leur utilité par rapport à l'utilité des autres : on peut ainsi donner de manière explicite l'avantage aux agents les plus pauvres ou les plus riches, ou encore par exemple concentrer l'importance sur les agents situés au milieu de l'échelle des richesses.

Définition 1.35 (Famille OWA) *La famille de fonctions d'utilité collective* moyenne pondérée ordonnée *(ou OWA pour Ordered Weighted Average) est la famille :*

$$g^{\overrightarrow{w}} = \sum_{i=1}^{n} w_i \times u_i^{\uparrow}, \text{ avec } \overrightarrow{w} \in [0,1]^n \text{ et } \sum_{i=1}^{n} w_i = 1.$$

Les deux fonctions d'utilité classiques admettent une représentation sous forme d'OWA : la fonction utilitariste correspond à l'OWA $g^{(\frac{1}{n},\dots,\frac{1}{n})}$, et la fonction égalitariste correspond à l'OWA $g^{(1,0,\dots,0)}$. D'autres fonctions classiques, mais moins utilisées dans le cadre de la décision collective

[8] Notons que pour une fonction d'utilité collective respectant le principe d'unanimité, cette propriété est équivalente à la concavité des courbes iso-utilités.

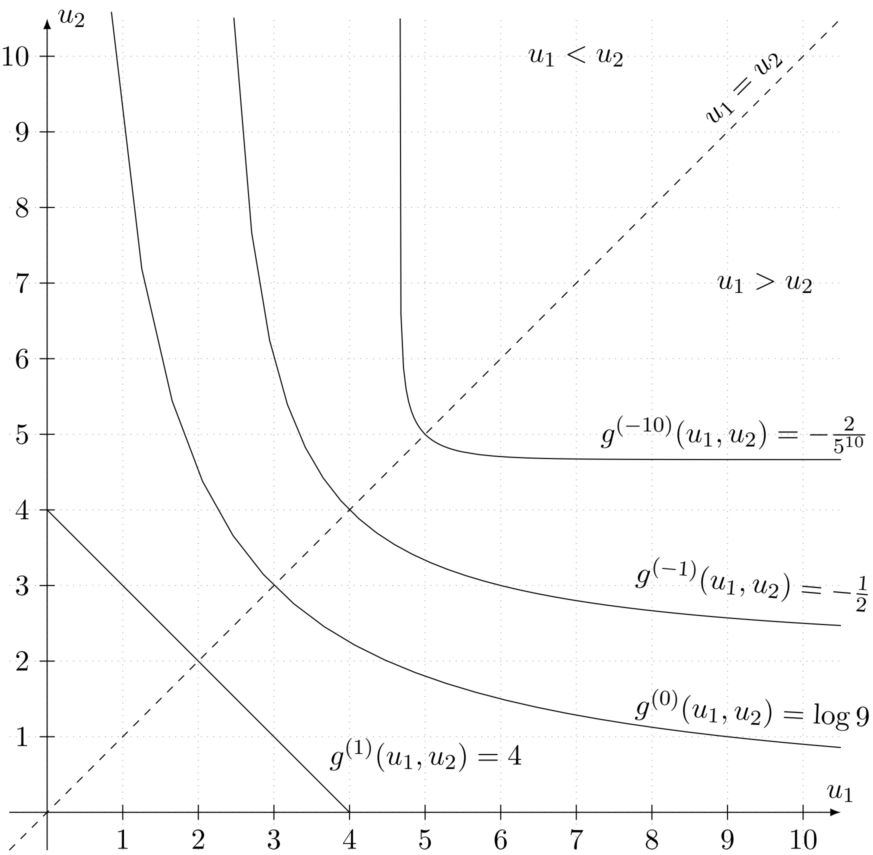


Figure 1.8 — Courbes iso-utilité collective de 4 fonctions d'utilité collective de la famille somme des puissances, pour 2 agents : $g^{(1)}$, $g^{(0)}$, $g^{(-1)}$ et $g^{(-10)}$.

équitable car peu appropriées dans ce contexte, peuvent être représentées par les moyennes pondérées ordonnées. Nous pouvons citer la fonction élitiste $g^{el} = \overrightarrow{u} \mapsto \max_i(u_i) = g^{(0,0,\ldots,1)}$ (qui est à l'opposé de la fonction égalitariste, mais peut être utile dans certains contextes où l'on s'intéresse à la maximisation de l'utilité d'un seul agent), ou encore la fonction dictatoriale de rang k, qui s'écrit $g^{\overrightarrow{w}}$, avec $w_i = 0$ si $i \neq k$, et $w_k = 1$, et généralise les fonctions égalitariste et élitiste. En outre, comme toutes les fonctions d'utilité collective introduites jusqu'ici, toutes les fonctions de la famille OWA vérifient la propriété d'insensibilité à une dilatation linéaire commune des utilités. Il existe de plus une caractérisation très simple des fonctions d'utilité collective de la famille OWA qui respectent le principe de réduction des inégalités : ce sont exactement les fonctions telles que $\forall(i, j),\ i > j \Rightarrow w_i < w_j$.

De plus, tout comme avec la famille somme des puissances, l'ordre leximin peut être représenté par un OWA si l'ensemble des utilités est fini ([Dubois *et al.*, 2001]) :

Proposition 1.5 *Si l'ensemble des profils d'utilité est fini, alors il existe un OWA qui représente l'ordre leximin.*

La preuve de cette proposition est détaillée à la section B.2.1 de l'annexe B.

Comme nous l'avons fait remarquer, la famille des moyennes pondérées ordonnées est construite pour permettre de contrôler précisément l'avantage donné aux faibles ou aux larges utilités dans l'agrégation, par la modulation des poids du vecteur $\overrightarrow{w}$. Ces notions d'avantage aux faibles ou aux larges utilités peut même être mesuré par des indices numériques : ainsi, Yager propose deux mesures caractéristiques du vecteur $\overrightarrow{w}$ [Yager, 1988] :

1. $\alpha(\overrightarrow{w}) = \sum_{i=1}^{n} \frac{n-j}{n-1} w_j$. Le coefficient $\alpha \in [0,1]$ mesure l'avantage donné aux utilités les plus faibles par rapport aux utilités les plus fortes. Par exemple, si $\overrightarrow{w} = (1, 0, \ldots, 0)$ (égalitarisme), $\alpha = 1$; si $\overrightarrow{w} = (\frac{1}{n}, \ldots, \frac{1}{n})$ (utilitarisme), $\alpha = 0.5$; si $\overrightarrow{w} = (0, 0, \ldots, 1)$ (élitisme), $\alpha = 0$.
2. $disp(\overrightarrow{w}) = -\sum_{i=1}^{n} \log(w_j) w_j$ (avec la convention $w \log(w) = 0$ si $w = 0$). Le coefficient $disp \in [0, \log(n)]$ mesure le degré d'utilisation de l'information contenue dans le vecteur d'utilités : si une seule valeur est utilisée, $disp = 0$; si l'OWA est symétrique, $disp = \log(n)$.

Enfin, la figure 1.9 montre un exemple de courbes iso-utilité collective de quatre fonctions de la famille OWA n'augmentant pas les inégalités pour un problème à deux agents, en fonction de l'utilité individuelle de ces deux agents, afin d'illustrer, tout comme pour la famille somme des puissances, la manière dont le vecteur de poids permet de moduler la fonction d'agrégation entre l'égalitarisme pur et l'utilitarisme.

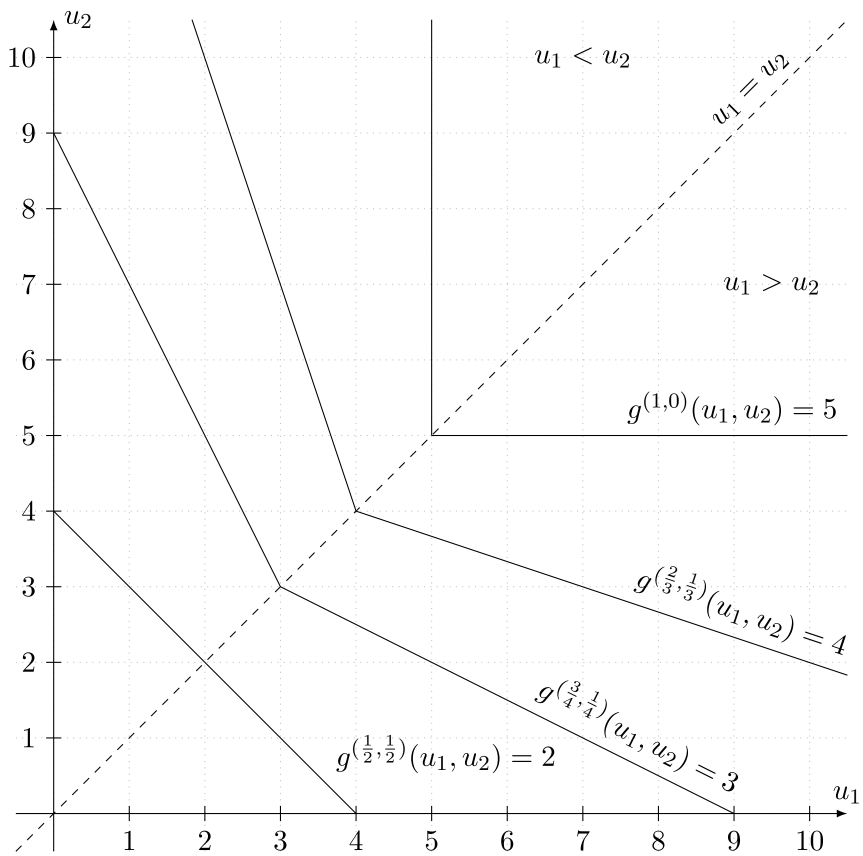


Figure 1.9 — Courbes iso-utilité collective de 4 fonctions d'utilité collective de la famille OWA, pour 2 agents : $g^{(1,0)}$, $g^{(\frac{2}{3},\frac{1}{3})}$, $g^{(\frac{3}{4},\frac{1}{4})}$ et $g^{(\frac{1}{2},\frac{1}{2})}$.

1.3.4.5 Normalisation des utilités

Il est un point crucial pour le bon fonctionnement du modèle *welfariste* cardinal que nous avons passé sous silence jusqu'ici : celui de la normalisation des utilités. Nous avons supposé implicitement dans la présentation du modèle que les utilités des agents étaient exprimées sur une échelle commune (monétaire par exemple), et donc que leur comparaison interpersonnelle était possible. Cependant, dans de nombreux problèmes réels, il est difficile d'imposer aux agents une échelle commune d'expression des utilités, et leur permettre de les exprimer librement empêche le bon fonctionnement de la justice sociale apportée par le modèle *welfariste* (sauf à utiliser la fonction de Nash qui est insensible à l'échelle individuelle des utilités) : par exemple, un agent qui sait que la fonction d'utilité collective utilisée est la fonction min peut très bien choisir de diviser sa fonction d'utilité individuelle par 10 pour se faire paraître plus malheureux qu'il n'est.

Deux solutions simples à ce problème existent. La première est d'imposer les échelles individuelles d'utilité, par exemple en attribuant le même nombre de points à chaque agent à répartir entre toutes les alternatives possibles. Toutefois, dans certains cas, une telle contrainte est trop contraignante pour l'expression des utilités. Dans ce cas, une normalisation des utilités individuelles avant partage est souhaitable. La solution classiquement adoptée est celle de Kalai-Smorodinsky [Kalai et Smorodinsky, 1975], qui normalise les utilités individuelles selon l'utilité maximale pouvant être obtenues par chaque agent s'il était seul dans le partage :

Définition 1.36 (Fonction d'utilité normalisée de Kalai-Smorodinsky) *Soit g une fonction d'utilité collective. Alors la* fonction d'utilité normalisée de Kalai-Smorodinsky *est la fonction :*

$$g^{(KS)} : (u_1, \ldots, u_n) \mapsto g(\frac{u_1}{\widehat{u_1}}, \ldots, \frac{u_n}{\widehat{u_n}}), \text{ où } \forall i, \widehat{u_i} = \max_{\overrightarrow{\pi} \in \mathscr{A}} u_i(\overrightarrow{\pi}).$$

En fait, la solution de Kalai-Smorodinsky est plus spécifique que cette définition, puisqu'elle impose aussi la fonction g à être la fonction égalitariste min : il s'agit au final d'un «égalitarisme relatif». Cette définition, comme nous l'avons fait remarquer plus haut, sort légèrement du cadre du *welfarisme*, puisque l'utilité collective dépend non seulement de l'ensemble des utilités individuelles, mais aussi de l'ensemble des utilités individuelles possibles : en d'autres termes, il s'agit d'une fonction de choix social, dans le cadre proposé dans [Nash, 1950].

Le problème de la normalisation des utilités pourrait constituer un vaste sujet d'étude, à mettre en relation avec l'étude des stratégies et de la manipulation. Nous avons choisi de ne pas aborder ces sujets, et nous considérerons donc par la suite que nous avons affaire à des utilités individuelles implicitement normalisées, et en tout cas commensurables.

1.4 Distribution ou répétition dans le temps de la procédure d'allocation

Si nous nous sommes attachés à décrire jusqu'ici l'ensemble des composantes permettant de modéliser un problème de partage, nous n'avons en revanche rien dit concernant la procédure d'allocation elle-même.

1.4.1 Partage centralisé ou distribué

La question de savoir qui partage la ressource, si elle n'est que peu pertinente dans le cas où on ne s'intéresse qu'à l'aspect qualitatif du partage (formalisé par le bien-être social), est en revanche un point crucial dans la mise au point de protocoles de partage. On peut distinguer deux grandes orientations : le partage centralisé ou le partage distribué.

Dans le premier cas, un agent particulier joue le rôle de distributeur de la ressource. Les agents se contentent de lui communiquer leurs préférences, sous la forme d'un protocole de partage prédéfini. L'agent distributeur a pour rôle d'allouer la ressource aux agents au vu des préférences qu'ils lui ont communiquées. Cette solution est la plus étudiée en informatique, notamment parce qu'elle a l'avantage de limiter les coûts de communication (dans le domaine des enchères électroniques par exemple, ce point est crucial), et parce que le pas en avant récent du domaine très étudié des enchères combinatoires [Cramton *et al.*, 2006; Sandholm, 1999, 2002; Rothkopf *et al.*, 1998; Lehmann *et al.*, 1999] a permis le développement et l'utilisation d'un ensemble d'algorithmes centralisés extrêmement performants, jouant en l'occurrence le rôle du commissaire-priseur des enchères. Les notions d'algorithmique sont à la base de la vision centralisée des problèmes d'allocation de ressource.

Le principal argument contre cette approche centralisée des problèmes de partage est que dans la réalité il peut être difficile de trouver un agent qui assume le rôle du «dictateur bienveillant» en charge du partage de la ressource, que ce soit pour des raisons de capacité de calcul (la vision centralisée a des besoins de calcul relativement conséquents) ou plus simplement pour des raisons d'absence de confiance en cet agent [Chevaleyre *et al.*, 2006a], ou en d'autres termes de confidentialité. La vision distribuée du problème de partage apparaît donc comme une alternative naturelle et intéressante lorsque l'on a affaire a des problèmes intraitables algorithmiquement, mais pour lesquels de simples petites améliorations par rapport à l'allocation initiale sont considérées comme des succès conséquents.

Dans le cas d'une approche distribuée de l'allocation, tous les agents jouent le même rôle. À partir d'un partage initial (supposé peu intéressant), les agents procèdent par échange d'objets pour arriver à une allocation supposée meilleure. Un tel échange est appelé *négociation* dans la littérature, bien que ce terme puisse prêter à confusion, puisqu'en fait, les agents n'ont que peu de latitude dans le choix des objets à échanger, et réagissent à un protocole bien déterminé. Les questions soulevées par cette approche sont diverses. Elles concernent par exemple la mise au point de protocoles de négociation et de dialogue entre les agents [Smith, 1980] ou, comme l'ont mis en avant certains travaux récents que nous citons plus loin, les propriétés de convergence vers un partage optimal, selon le type d'échanges autorisés. Ces derniers travaux se sont concentrés sur des échanges *rationnels* avec compensation monétaire possible, c'est-à-dire tels qu'à la fin de l'échange, toute baisse éventuelle d'utilité d'un agent est compensée par une somme d'argent valant au moins l'utilité perdue. Cette hypothèse est raisonnable : dans un cadre réel, un agent humain n'accepte de procéder à un échange que s'il a quelque chose à y gagner (l'altruisme pur est exclu de ce genre de problèmes) : soit un objet, soit une compensation monétaire. En général, ces travaux se concentrent aussi sur des échanges très simples (c'est-à-dire impliquant un nombre limité d'agents et d'objets) : les plus simples de ces échanges sont des 1-*deals*, impliquant un seul objet (ainsi que la monnaie utilisée pour le paiement) et deux agents.

De manière très intéressante, les propriétés de convergence de séquences d'échanges vers un optimum global sont nombreuses [Chevaleyre *et al.*, 2005a,b; Endriss *et al.*, 2006; Chevaleyre *et al.*, 2007a]. La première d'entre elles est due à Sandholm [Sandholm, 1998] : toute séquence d'échanges rationnels (avec compensation monétaire) est finie et converge vers un optimum utilitariste. Ce résultat, s'il est intéressant d'un point de vue théorique, a cependant peu d'application pratique. En effet, de manière générale il se peut qu'à un stade de la négociation il n'existe que des échanges rationnels extrêmement compliqués, et en particulier impliquant de nombreux agents et objets.

Il existe cependant des résultats concernant les séquences d'échanges simples (1-*deals*), pour peu que l'on ajoute quelques hypothèses à la forme des fonctions d'utilité des agents. Ainsi, si les utilités des agents sont additives, toute séquence d'échanges simples rationnels est finie et conduit à un optimum utilitariste [Endriss *et al.*, 2006]. D'autres types de restrictions (échanges coopératifs, échanges égalitaires) ont aussi été étudiées. Ces restrictions conduisent à d'autres types de résultats de convergence (convergence vers une allocation Pareto-optimale, convergence vers une allocation optimale au sens égalitariste, ...). Parallèlement a été étudié le lien entre certaines classes de fonctions d'utilité et les propriétés de convergence : quelques classes de fonctions d'utilité garantissant la convergence vers un optimum utilitariste des séquences d'échanges rationnels pour un certain type de *deals* ont été mises en évidence [Chevaleyre *et al.*, 2005b].

Si le lien entre optimisation d'une fonction d'utilité et échange rationnel est assez clair, et fournit donc un ensemble de résultats de convergence assez intuitifs, le lien entre ces échanges rationnels et l'absence d'envie est plus difficile à obtenir : d'une part l'absence d'envie est fondée sur l'appréciation personnelle d'une situation, et d'autre part le mécanisme d'échange rationnel d'objets, même s'il s'agit d'un mécanisme local, tend à faire augmenter l'utilité collective. Des travaux récents, présentés dans [Chevaleyre *et al.*, 2007a], établissent un lien entre l'absence d'envie et les processus de négociation rationnelle : moyennant quelques hypothèses sur les fonctions d'utilité (additivité, superadditivité), sur les fonctions de paiement qui imposent la valeur des compensations monétaires aux agents en fonction des utilités des objets échangés, ou encore sur l'allocation initiale, il est possible de garantir la convergence de toute séquence d'échanges vers un état Pareto-efficace et sans envie (notons qu'en présence de compensations monétaires et de propriétés de superadditivité sur les fonctions d'utilité individuelle il en existe toujours un). Ces travaux ont aussi conduit à la proposition de mesures d'envie et à l'étude expérimentale de la rapidité de convergence vers un état à envie minimale par une séquence d'échanges rationnels. L'ensemble de ces travaux est présenté dans la thèse [Estivie, 2006].

Notons enfin qu'il existe un ensemble de travaux relativement récents sur la distribution de la résolution d'un problème d'optimisation. Ces travaux, qui ont une portée beaucoup plus générale que le simple problème de partage, mais s'y appliquent parfaitement, portent sur le développement d'algorithmes de résolution décentralisés appliqués à des problèmes d'optimisation combinatoire impliquant un certain nombre d'agents, nombre potentiellement inconnu ou non borné. Ce genre de problèmes peut se trouver par exemple dans le domaine des fournisseurs de services en ligne sur Internet, dont le rôle est de proposer à un ensemble (inconnu) de clients un certain nombre de services proposés par un ensemble (potentiellement large) de fournisseurs. Il est impensable dans ce contexte d'envisager une modélisation et une résolution centralisée du problème, d'autant plus que les variables d'un tel problème sont susceptibles d'évoluer dans le temps.

L'idée à la base de tous les travaux de recherche sur le sujet des problèmes d'optimisation décentralisés est de déléguer aux agents la résolution des sous-problèmes locaux qui les concernent. La solution partielle calculée par ces agents est ensuite propagée de manière synchrone ou asynchrone sous la forme de messages. Ces techniques sont appliquées avec succès à la résolution décentralisée de problèmes de choix social, et les derniers travaux sur le sujet intègrent des techniques de résistance aux manipulations. On pourra consulter [Faltings, 2006] pour avoir un aperçu détaillé de ces techniques, et on pourra trouver dans [Petcu *et al.*, 2006] un exemple d'algorithme de résolution d'un problème d'optimisation distribué fondé sur la délégation de la résolution aux agents, et sur un mécanisme de résistance aux manipulations.

Par la suite, nous laisserons de côté les problèmes liés au partage distribué, afin de nous concentrer uniquement sur les procédures d'allocation centralisées.

1.4.2 Répétition dans le temps du problème d'allocation

Nous conclurons notre taxonomie des problèmes de partage en évoquant le sujet de la répétition des problèmes de partage dans le temps. Les applications du monde réel ne se résument généralement pas à un partage unique. Au lieu de cela, elles impliquent souvent un ensemble de partages distincts, ou un partage répété dans le temps. L'application 1 concernant la constellation Pléiades est un exemple parfait : chaque jour le centre de programmation et planification doit réaliser le partage des prises de vue entre les agents.

Il faut dans ce cas reconsidérer notre modèle. Là où un partage unique force les agents (et le décideur) à un grand nombre de concessions en vertu du principe d'équité, ces concessions peuvent être évitées ou atténuées dans le cas d'un partage répété dans le temps, car l'équité peut être obtenue dans ce cas par *régulation temporelle* : l'inéquité d'un partage à l'instant t peut être compensée par l'inéquité d'un autre partage à un instant $t' > t$. Il serait donc dommage dans ce cas de vouloir traiter chaque occurrence du problème de partage séparément avec le modèle *welfariste* introduit ci-avant, car on perdrait l'ensemble des bénéfices dus à la régulation temporelle. Bien entendu, si le nombre de répétitions du partage est fini et connu à l'avance, on peut traiter l'ensemble des instances comme un problème global, mais on se heurte alors à plusieurs écueils :

- ▷ celui de l'explosion combinatoire ;
- ▷ celui de connaître à l'avance le nombre d'occurrences du problème de partage dans le temps (dans le cas du problème Pléiades, on peut même considérer en première approximation qu'il est infini), ainsi que les instances elles-mêmes ;
- ▷ celui des préférences des agents, qui sont souvent dépendantes du temps.

Alors qu'il existe une grande littérature sur le problème relativement voisin des *jeux répétés* [Aumann et Hart, 2002], en revanche, les problèmes d'allocation répétée n'ont été que peu étudiés à notre connaissance. On compte toutefois une exception à cette remarque, constituée par l'ensemble des travaux sur le problème Pléiades, notamment [Lemaître *et al.*, 1999, 2004], qui proposent un ensemble de modèles et protocoles permettant de prendre en compte la régulation temporelle dans la recherche de solutions efficaces et équitables. Le moyen proposé dans ces travaux pour prendre en compte la régulation temporelle est de traiter le problème de partage à l'instant t en intégrant les variables de k problèmes de partage antérieurs à t en tant que données figées mais influençant le partage courant. En d'autres termes, on résout à chaque temps t un problème constitué du problème courant et des k derniers problèmes résolus. Ce mécanisme de fenêtre glissante permet d'assurer l'équité et l'efficacité en s'appuyant sur la régulation temporelle.

1.5 Conclusion

Nous avons tâché tout au long de ce chapitre de dresser une taxonomie succincte des problèmes de partage, articulée autour de la modélisation des briques de base de ces problèmes. Nous nous sommes concentrés sur les aspects suivants, inspirés de l'article [Chevaleyre *et al.*, 2006a] :

- ▷ Le type de la ressource : elle peut être continue, discrète, indivisible, multi-unités.
- ▷ La modélisation des préférences : les préférences des agents peuvent être représentées par une structure ordinale générale ou particulière (dichotomique, ordres d'intervalles, semi-ordres), qualitative, ou numérique.
- ▷ L'agrégation des préférences et les propriétés des partages : le modèle dominant dans le domaine du partage est le *welfarisme* cardinal, fondé sur les ordres de bien-être social portant sur les profils d'utilité : ordres utilitariste, égalitariste, leximin, somme des puissances, OWA… Un certain nombre de propriétés permettent de caractériser les partages, décisions collectifs ou ordres de bien-être sociaux : Pareto-efficacité, anonymat, juste part, absence d'envie, réduction des inégalités, …
- ▷ La procédure d'allocation : celle-ci peut être centralisée ou distribuée entre les agents.
- ▷ La répétition dans le temps : on peut avoir affaire à un problème de partage simple ou répété dans le temps. Les modèles entrant en jeu ne sont pas les mêmes.

Bien entendu, cette introduction n'a pas la prétention ni la vocation d'être exhaustive, et certains problèmes relatifs au partage, ou certains aspects ont été éludés dans ce chapitre. Ainsi, nous aurions pu évoquer d'autres sujets aussi divers que l'introduction d'incertitudes dans le partage, la prise en compte d'informations incomplètes, ou encore la notion de stratégies de manipulation. Nous renvoyons le lecteur intéressé à la littérature abondante sur ces sujets.

Nous allons maintenant définir les bornes de notre étude des problèmes de partage, en nous inspirant de l'ensemble des considérations introduites dans ce chapitre.

Tout d'abord, nous nous limiterons au partage de biens *indivisibles*, et sans compensation monétaire. Cela exclut d'emblée tous les problèmes du type partage de territoires, d'investissements financiers, ou encore de biens après divorce ou décès si les compensations monétaires sont autorisées. Nous nous autorisons n'importe quel type de contraintes sur l'espace des allocations. Ensuite, nous nous plaçons dans le cadre du *welfarisme* cardinal, pour lequel les préférences des agents sont représentées par des indices numériques. Nous nous intéresserons de manière générale à tous les critères et fonctions d'utilité collective introduits lors de la présentation de ce modèle. Enfin, nous nous limiterons à un partage centralisé et non répété dans le temps.

Définition 1.37 (Instance du problème de partage de biens indivisibles) *Une instance du problème de partage de biens indivisibles est un tuple* $(\mathscr{N}, \mathscr{O}, \mathscr{C}, (f_1, \ldots, f_n), \succeq, \mathscr{V})$*, où :*

- ▷ $\mathscr{N} = \{1, \ldots, n\}$ *est un ensemble fini d'agents ;*
- ▷ $\mathscr{O}$ *un ensemble fini d'objets ;*
- ▷ $\mathscr{C}$ *est un ensemble de contraintes d'admissibilité, c'est-à-dire de sous-ensembles de* $\mathscr{O}^n$ *;*
- ▷ $(f_1, \ldots, f_n)$ *est un ensemble de fonctions d'utilité :* f_i *est la fonction d'utilité de l'agent* i*, qui à toute part* $\pi_i \subseteq \mathscr{O}$ *associe une utilité* $f_i(\pi_i) \in \mathscr{V}$ *;*
- ▷ $\succeq$ *est un ordre de bien-être social sur l'ensemble des profils d'utilité admissibles, c'est-à-dire sur l'ensemble des profils vérifiant toutes les contraintes, éventuellement défini par une fonction d'utilité collective.*

Une solution du problème de partage de biens indivisibles est un partage admissible. Une solution optimale de ce problème est un partage admissible dont le profil d'utilité associé est non dominé au sens de l'ordre de bien-être social $\succeq$. Par la suite, nous nous intéresserons non seulement aux partages optimaux au sens de $\succeq$, mais aussi à l'existence de partages vérifiant certaines propriétés comme l'absence d'envie et la Pareto-efficacité (voir chapitre 4).

Dans le chapitre 2, nous allons introduire une extension de ce modèle afin de prendre en compte la notion de droits exogènes inégaux dans le partage. Puis nous introduirons dans le chapitre 3 un modèle formel du problème de partage de biens indivisibles, qui spécifiera notamment la manière dont seront exprimées les contraintes et les préférences.

Chapitre 2

Droits exogènes

Nous avons introduit au chapitre 1 l'ensemble des éléments formant la base de la modélisation des problèmes de partage. Parmi ces éléments, nous nous sommes en particulier intéressés à la manière dont les préférences des agents peuvent être agrégées pour aboutir finalement à une décision collective (c'est-à-dire dans notre cas à un partage) équitable. Comme nous l'avons vu, la théorie du *welfarisme* cardinal fournit un modèle particulièrement intéressant du problème d'agrégation de préférences individuelles, modèle fondé sur une représentation cardinale des préférences et une agrégation par le biais d'un ordre de bien-être social ou d'une fonction d'utilité collective. Il permet, dans le contexte des problèmes de partage, d'explorer tout un ensemble de compromis entre le principe de compensation — correspondant à l'égalitarisme pur — et le principe d'adéquation (ou *sum-fitness*) — correspondant à l'utilitarisme classique. En revanche, toute idée de récompense est absente de ce modèle.

Intéressons-nous maintenant au principe de droits exogènes. Ce principe permet de prendre en compte dans le problème d'allocation des considérations complètement extérieures à ce problème. Ces raisons extérieures peuvent être diverses, comme nous le verrons sur les exemples ci-après, mais ne peuvent pas provenir directement de la contribution à la création de la ressource, car dans ce cas le principe de droits exogènes interfère avec le principe de récompense (la ressource à l'agent le plus méritant). Ce principe de droits exogènes est présent de manière implicite dans le modèle *welfariste* sous la forme de la propriété d'*anonymat*[1], s'exprimant par la symétrie de l'ordre de bien-être social. Dans ce modèle, les agents sont placés sur un pied d'égalité vis-à-vis de la procédure d'allocation : en d'autres termes, cette procédure est complètement aveugle à l'identité des agents.

Cependant, de nombreux problèmes réels de décision collective et en particulier de partage impliquent des agents qui ne sont pas égaux vis-à-vis de la procédure de décision, et ce pour des raisons aussi diverses que celles exposées dans les exemples ci-après. Nous traduirons cette différence d'importance entre agents par la notion de *droits des agents*, ces droits étant représentés par des indices numériques : plus le droit est élevé, plus l'agent est censé bénéficier de la décision, d'une manière que nous cherchons à capturer précisément.

Voici quelques exemples de problèmes réels nécessitant la prise en compte de droits inégaux sur le partage :

- ▷ **Populations de tailles différentes :** La ressource est un bien de consommation, et les agents sont des populations de tailles différentes. Les agents ont des besoins différents, reconnus par le groupe. Le droit d'une population est sa taille.
- ▷ **Constitution de comité :** Les agents sont des circonscriptions de tailles différentes. On constitue un comité de représentants des circonscriptions. Le nombre total de représentants

[1] Les droits sont égaux dans ce cas.

est fixé. Il s'agit de décider du nombre de représentants par circonscription. Les droits sont les tailles (en nombre d'habitants) des circonscriptions.

- ▷ **Banqueroute :** Une société en faillite doit être liquidée. Son actif — dont la valeur est inférieure à la somme des dettes — est réparti entre ses créanciers, qui doivent décider de cette répartition. Le droit d'un créancier est le montant de sa créance.
- ▷ **Ressource commune avec investissement initial :** Une ressource commune est exploitée par un groupe d'agents. Ces agents ont participé différemment à la production initiale de la ressource, et à ce titre, ont des revendications particulières sur son exploitation. Chaque agent attend bien entendu un retour sur investissement en rapport avec celui-ci. Cet exemple est illustré parfaitement par l'application 1 concernant l'allocation de prises de vues pour la constellation de satellites Pléiades. Ici, le droit d'un agent correspond à son investissement initial.
- ▷ **Productivités différentes :** Un groupe d'agents exploite une ressource commune et la transforme en biens revendus au bénéfice du groupe. Certains sont plus productifs que d'autres. Quelle est la meilleure allocation de la ressource aux agents ? Le droit d'un agent est sa productivité[2].

Les fonctions d'utilité collective et ordres de bien-être social introduites dans le cadre du *welfarisme* cardinal et décrites dans le chapitre 1 sont incapables de prendre en compte une inégalité des droits exogènes, car elles vérifient toutes le principe élémentaire d'anonymat. Nous allons donc proposer une extension du cadre *welfariste* qui va permettre d'intégrer l'inégalité des droits exogènes dans le modèle. Notre démarche est la suivante : nous allons dans un premier temps étendre les définitions d'ordre de bien-être social et de fonction d'utilité collective afin de prendre en compte les droits des agents. Puis nous introduirons une méthode permettant de bâtir des ordres de bien-être social et fonctions d'utilité collective à droits inégaux à partir de leurs équivalents classiques et d'un vecteur de droits des agents. Enfin, nous proposerons une extension des propriétés classiques introduites à la section 1.3.3, et des fonctions d'utilité collectives classiques introduites à la section 1.3.4, ainsi que l'introduction de nouvelles propriétés liées aux droits inégaux. Cette démarche sera illustrée de nombreux exemples.

Notons, avant d'aborder le sujet des droits inégaux, que ce problème n'est pas sans rappeler un domaine d'étude relativement similaire : celui des indices de pouvoir [Felsenthal et Machover, 1998] en théorie du vote. Le contexte est similaire, mais les questions posées sont différentes : dans le domaine du vote, les principaux travaux sur les indices du pouvoir ne concernent pas la mise au point de la procédure de vote (contrairement à ce que nous cherchons à faire en intégrant les droits exogènes au modèle *welfariste* cardinal), mais plutôt la question de l'attribution des poids aux votants ou encore l'analyse de leur influence sur le résultat du vote. Ce sont par exemple des questions cruciales dans le cadre du conseil de l'Union Européenne, où la répartition des poids de vote aux différents pays suscite un certain nombre de polémiques houleuses.

On peut se demander pourquoi le problème de la généralisation des fonctions d'utilité collective à des droits inégaux n'est pas autant étudié dans la littérature sur le *welfarisme* qu'il ne l'est dans le domaine du vote. La raison que nous voyons à cela est que souvent les problèmes de décision à droits inégaux sont complexes et particuliers, et font l'objet de modèles et travaux dédiés à chacun. C'est le cas des exemples de la constitution de comités, de la banqueroute, des productivités différentes, qui sont des exemples classiques étudiés en tant que tels. Notre démarche se veut plus générale : nous recherchons un cadre permettant d'aborder de manière unifiée ce type de problèmes.

[2]Cet exemple se situe à la limite du cadre des droits exogènes, dans la mesure où la «productivité» d'un agent constitue plutôt une propriété intrinsèque de sa fonction d'utilité, et qu'il interfère avec le principe de récompense. Nous conservons tout de même l'exemple car il entre formellement dans le schéma méthodique proposé et son traitement dans ce cadre apporte une solution tout-à-fait plausible.

2.1 Exemples repères

Par une suite d'exemples, nous allons montrer comment, à partir d'un même contexte d'allocation de ressource entre agents, différentes fonctions d'utilité collective peuvent se justifier. Ces exemples nous permettront d'introduire les droits inégaux de manière concrète. Le contexte est le suivant : une collectivité de fermiers utilise un système commun de distribution d'eau captée. Ils doivent décider ensemble de la quantité d'eau allouée annuellement à chaque fermier, sachant que toutes les distributions ne sont pas admissibles : la quantité d'eau disponible est limitée, les tuyaux sont plus ou moins gros... En fonction de l'attribution annuelle d'eau $(a_1, \dots a_i, \dots a_n)$, le fermier i retire une utilité individuelle $u_i = f_i(a_i)$. Cette fonction est propre à chaque fermier car ils ne cultivent pas tous les mêmes plantes, les sols n'ont pas le même rendement, l'exposition au soleil est différente... On suppose que les fermiers ont une capacité de travail identique. On admettra qu'il existe une échelle commune des utilités, et qu'elles sont mesurées par exemple en euros. Nous allons d'abord supposer que nos agents ont des droits égaux, ce qui nous permettra de rappeler les deux fonctions d'utilité collectives les plus importantes introduites au chapitre 1, puis nous examinerons le cas de droits inégaux.

2.1.1 Avec des droits égaux

1. Utilitarisme classique

La collectivité s'intéresse à l'utilité globale produite. La fonction d'utilité collective naturelle est celle qui correspond à la vision utilitariste classique :

$$g^{\star}(\overrightarrow{u}) = \sum_{i=1}^{n} u_i$$

On note l'aspect interchangeable des utilités individuelles : pour atteindre un haut niveau d'utilité collective, un bas niveau d'utilité produite par une ferme devra être compensé par un bon niveau d'une autre. Un litre d'eau supplémentaire ira au fermier ayant la plus grande utilité individuelle marginale (en supposant que l'utilité individuelle, fonction de la quantité d'eau reçue n'est pas décroissante, et à condition que les contraintes d'admissibilité soient satisfaites). Si l'utilité individuelle représente ou est connectée directement au salaire du fermier, le choix de cette fonction d'utilité collective implique qu'un fermier sera amené à se sacrifier pour la communauté.

2. Égalitarisme

L'utilité individuelle représente le revenu du fermier. La collectivité s'intéresse maintenant à une répartition égalitaire de l'utilité produite par chaque fermier. Une fonction qui convient est celle qui correspond à la vision égalitariste du partage,

$$g^{(e)}(\overrightarrow{u}) = \min_{i=1}^{n} u_i$$

car elle tend à la fois à égaliser les revenus et à les tirer vers le haut. Noter l'absence totale d'interchangeabilité des utilités individuelles : un litre d'eau supplémentaire ira au fermier possédant l'utilité la plus faible, même si ce litre est plus productif en utilité dans une autre ferme. Bien entendu, on pourra utiliser le préordre leximin en lieu et place de la fonction d'utilité collective min pour pallier dans une certaine mesure ses inconvénients.

2.1.2 Avec des droits inégaux

Considérons maintenant que les fermes sont habitées par des familles de tailles inégales. La taille de la famille i sera notée e_i. Il faut en tenir compte dans le choix de l'allocation.

1. Division avec utilitarisme classique

L'utilité représente le revenu de la ferme. Elle est divisée entre chaque habitant individuellement. Chaque habitant reçoit donc u_i/e_i. La collectivité s'intéresse au bien-être collectif mesuré par la somme de ce que reçoivent tous les habitants. La fonction d'utilité collective est donc :

$$g^{\star}_{\overrightarrow{e}}(\overrightarrow{u}) = \sum_{i=1}^{n}(e_i \cdot (u_i/e_i)) = \sum_{i} u_i$$

2. Division avec égalitarisme

Même cas de figure que dans l'exemple précédent, mais la collectivité s'intéresse à une répartition équitable de l'utilité reçue par chacun des habitants individuellement. Une fonction d'utilité collective convenable est alors :

$$g^{(e)}_{\overrightarrow{e}}(\overrightarrow{u}) = \min_{i=1}^{n}(u_i/e_i)$$

3. Indivision avec utilitarisme classique

On change maintenant de point de vue sur l'utilité individuelle. L'utilité u_i caractérise la «prospérité» de la ferme i en tant que propriété indivisible, et mesure en quelque sorte l'agrément d'y habiter. Chaque habitant d'une ferme profite de la prospérité de sa ferme et de celle-ci seulement. Les habitants d'une ferme en jouissent d'une manière équivalente. Chaque habitant de la ferme i reçoit donc l'utilité u_i de manière indivisible. Puis, la collectivité des fermiers s'intéresse à maximiser l'agrément total de tous les habitants, mesurée comme la somme des utilités reçues par les habitants. L'utilité collective est alors la somme pondérée :

$$g^{\star}_{\overrightarrow{e}}(\overrightarrow{u}) = \sum_{i=1}^{n}(e_i \cdot u_i)$$

4. Indivision avec égalitarisme

Le point de vue est le même que dans l'exemple précédent : l'utilité individuelle caractérise la «prospérité» de la propriété, et chaque habitant de la ferme i reçoit l'utilité u_i. La collectivité s'intéresse maintenant à une répartition équitable entre chacun des habitants individuellement. L'utilité collective convenable est

$$g^{(e)}_{\overrightarrow{e}}(\overrightarrow{u}) = \min_{i} u_i$$

Cette série d'exemples illustre la diversité des fonctions d'utilité collective possibles, selon le but poursuivi par la communauté et selon la nature des satisfactions des agents, lorsque les agents sont pourvus de droits inégaux. Dans la suite de l'article, nous chercherons à rendre compte de manière systématique de ces différentes formes, et éventuellement à en proposer de nouvelles.

2.2 Le principe de duplication des agents

Les quatre exemples précédents ont introduit la notion de fonction d'utilité collective à droits exogènes, et ont mis en valeur la façon intuitive de construire une telle fonction d'utilité : par duplication des agents en fonction de leurs droit exogène, et en calculant l'utilité individuelle de ces «sous-agents» à l'aide d'une certaine fonction. Nous allons maintenant introduire ce principe de manière formelle, sous le nom de «principe de duplication des agents». Il nous faut cependant tout d'abord définir ce que l'on entend par «droit exogène».

Définition 2.1 (Vecteur de droits exogènes) *Soit $\mathscr{N}$ un ensemble d'agents de taille n. Un* vecteur de droits exogènes *est un vecteur $\overrightarrow{e} \in \mathbb{N}_\star^n$.*

Nous nous restreignons donc à des droits exogènes définis par des entiers. Cette hypothèse n'est pas très contraignante en pratique. D'une part il est rare que l'on exprime de manière naturelle des droits par des nombres non entiers, et d'autre part, si les droits sont rationnels[3], la propriété d'insensibilité à une dilatation proportionnelle de l'échelle commune d'expression de ces droits, que nous allons introduire dans la section 2.3.3, nous autorisera à les rendre entiers en les multipliant par un multiple commun à tous leurs dénominateurs.

Par la suite, nous utiliserons aussi la version normalisée du vecteur de droits exogènes, que nous noterons $\overrightarrow{\underline{e}}$. Chaque composante de ce vecteur sera définie de la manière suivante :

$$\underline{e_i} = \frac{e_i}{m}, \text{ avec } m = \sum_{j=1}^{n} e_j.$$

Nous nous devons d'ajouter une précision de première importance concernant le vecteur de droits exogènes. Nous supposons que ce vecteur de droits exogènes a été défini de manière préalable au problème, que tous les agents sont d'accord sur ce vecteur, et qu'il n'est pas remis en cause durant la procédure d'allocation. En d'autres termes, la manière dont a été défini le vecteur ne nous concerne pas ici, mais nous supposons que sa définition est acceptée sans réserve par tous les agents. Cette hypothèse est très importante en particulier pour la définition de l'équité en présence de droits exogènes : les frustrations relatives à la définition des droits elle-même ne peuvent servir de base à la mesure de l'inéquité.

Le *principe de duplication* est un moyen de résoudre le problème de prise de décision collective en présence de droits inégaux. L'idée est de remplacer chaque agent par autant de *clones* qu'il possède de droits (ou d'un nombre de clones proportionnel à ses droits si les droits ne sont pas des entiers), l'utilité reçue par chaque agent étant répartie — d'une manière qui est discutée plus loin — entres ses clones. L'idée est ensuite de se ramener à un problème de décision collective entre les m clones considérés avec des droits égaux. Le raisonnement vise à conférer à chaque agent un «pouvoir de décision» égal ou proportionnel à son droit.

Il semble que ce principe a été introduit pour la première fois dans [Steinhaus, 1948] (cité par [Brams et Taylor, 1996]), et il est repris dans quelques travaux [Brams et Taylor, 1996; Pivato, 2006]. Cependant, il est toujours appliqué dans un contexte de division équitable de ressources, qui conduit à la fonction $\min_i u_i/e_i$, ce qui n'est pas toujours pertinent, comme nous l'avons vu dans les exemples précédents. Nous proposons donc une formalisation du principe de duplication, autorisant une utilisation plus large que celle qu'on lui donne habituellement et faisant intervenir deux paramètres :

▷ la manière dont l'utilité d'un agent est répartie entre ses clones,

▷ l'ordre de bien-être collectif jouant sur la société des clones.

Nous supposerons par la suite que les fonctions d'utilité sont définies sur un espace $\mathscr{V}$ qui correspond à l'espace des rationnels positifs $\mathbb{Q}^+$ ou à l'espace des réels positifs $\mathbb{R}^+$.

Le principe de duplication est fondé sur la notion *fonction de répartition.* Le rôle de cette fonction est de faire correspondre à l'utilité u_i d'un agent i et à son droit e_i l'utilité d'un de ses clones :

[3] Probablement aucune application ne nécessitera l'emploi de nombres réels irrationnels en guise de droits exogènes.

Définition 2.2 (Fonction de répartition) *Une* fonction de répartition *est une fonction* $\div : \mathscr{V} \times \mathbb{N}_\star \to \mathscr{V}$.

Deux fonctions de répartition sont naturelles. D'une part la division ordinaire $u \div e \stackrel{def}{=} u/e$, qui prend son sens dans le cas d'une satisfaction devant être nécessairement divisée entre les clones, c'est-à-dire lorsque les utilités individuelles sont *préemptives*, et d'autre part la simple réplication $u \div e \stackrel{def}{=} u$, qui convient dans le cas d'une satisfaction qui ne s'épuise pas lorsqu'elle est partagée (exemple de la «prospérité» dans les exemples précédents). On notera, pour tout agent i, r_i l'utilité d'un des clones correspondant à l'agent i, soit $r_i = u_i \div e_i$.

Définition 2.3 (Ordre de bien-être social à droits inégaux) *Soient $\mathscr{N}$ un ensemble de n agents, $\overrightarrow{e} \in \mathbb{N}_\star^n$ un vecteur de droits inégaux et $\succeq$ un ordre de bien-être social classique sur $\mathscr{V}^m$. L'*ordre de bien-être social à droits inégaux *$\succeq_{\overrightarrow{e}}$ correspondant à $\succeq$ et à $\overrightarrow{e}$ est défini par le principe de duplication des agents comme un préordre sur $\mathscr{V}^n$ tel que :*

$$\overrightarrow{u} \preceq_{\overrightarrow{e}} \overrightarrow{v} \Leftrightarrow \overrightarrow{u} \triangleright_{\overrightarrow{e}}^{\div} \preceq \overrightarrow{v} \triangleright_{\overrightarrow{e}}^{\div}\text{, avec } \overrightarrow{u} \triangleright_{\overrightarrow{e}}^{\div} \stackrel{def}{=} \big(\underbrace{r_1, \ldots, r_1}_{e_1 \text{ fois}}, \ldots, \underbrace{r_n, \ldots, r_n}_{e_n \text{ fois}}\big) \text{ et } r_i = u_i \div e_i.$$

De même, nous pouvons définir la notion de fonction d'utilité collective à droits inégaux selon le principe de duplication des agents

Définition 2.4 (Fonction d'utilité collective à droits inégaux) *Soient $\mathscr{N}$ un ensemble de n agents, $\overrightarrow{e} \in \mathbb{N}_\star^n$ un vecteur de droits inégaux et g une fonction d'utilité collective classique sur $\mathscr{V}^m$. La* fonction d'utilité collective à droits inégaux *$g_{\overrightarrow{e}}$ correspondant à g et à $\overrightarrow{e}$ est définie par le principe de duplication des agents de la manière suivante :*

$$g_{\overrightarrow{e}} : \begin{array}{ccc} \mathscr{V}^n & \to & \mathscr{V} \\ \overrightarrow{u} & \mapsto & g(\overrightarrow{u} \triangleright_{\overrightarrow{e}}^{\div}) \end{array}\text{, avec toujours } \overrightarrow{u} \triangleright_{\overrightarrow{e}}^{\div} \stackrel{def}{=} \big(\underbrace{r_1, \ldots, r_1}_{e_1 \text{ fois}}, \ldots, \underbrace{r_n, \ldots, r_n}_{e_n \text{ fois}}\big) \text{ et } r_i = u_i \div e_i.$$

Le processus de prise de décision collective est donc constitué de deux étapes successives : duplication des agents, puis agrégation des utilités des clones. Ce principe est illustré sur la figure 2.1.

Énumérons les quatre fonctions d'utilité collective à droits inégaux qui résultent des deux choix possibles introduits pour $\div$ et pour g dans les exemples repères :

	$u \div e \stackrel{def}{=} u/e$ (division)	$u \div e \stackrel{def}{=} u$ (réplication)
$g \stackrel{def}{=} \sum^{(m)}$ (utilitarisme classique)	$\sum_{i=1}^{n} u_i$	$\sum_{i=1}^{n} (e_i \cdot u_i)$
$g \stackrel{def}{=} \min^{(m)}$ (égalitarisme)	$\min_{i=1}^{n} (u_i/e_i)$	$\min_{i=1}^{n} u_i$

Nous avons indiqué d'un mot-clé la caractéristique importante de chaque fonction de répartition (division / réplication), et de même pour la fonction d'utilité collective sur les clones (utilitarisme classique / égalitarisme). Les notations $\sum^{(m)}$ et $\min^{(m)}$ rappellent que ces opérateurs portent sur m opérandes. On notera que la fonction d'utilité collective $\min_i(u_i/e_i)$ tend vers l'égalité des rapports u_i/e_i et donc vers la proportionnalité des utilités individuelles par rapport aux droits. On pourra bien entendu utiliser l'ordre social leximin appliqué aux m clones en lieu et place de la fonction min.

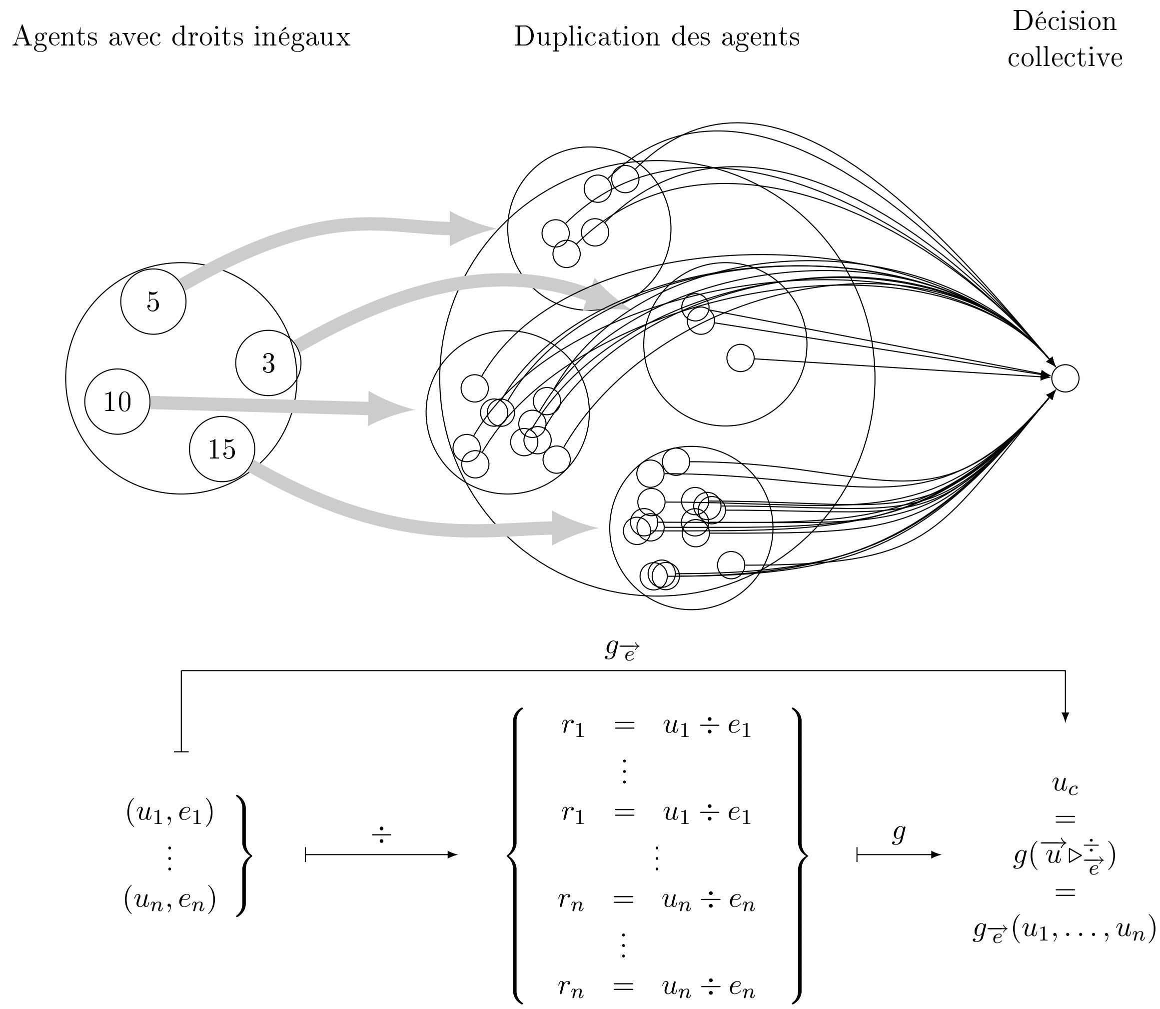


Figure 2.1 — Illustration du principe de duplication des agents.

2.3 Propriétés des ordres sociaux et partages optimaux

L'introduction de droits exogènes inégaux dans le domaine de la prise de décision collective modifie non seulement la notion de fonction d'utilité collective, mais aussi les propriétés raisonnables classiques qui permettent de caractériser ces fonctions d'utilité. Nous allons proposer dans un premier temps une extension des propriétés classiques caractérisant les ordres sociaux et les partages optimaux, afin qu'elles prennent en compte les droits exogènes inégaux. Nous nous intéresserons ensuite plus particulièrement aux propriétés d'équité, dont l'adaptation à ce contexte est légèrement plus délicate. Enfin, nous introduirons quelques nouvelles propriétés dont l'objectif est de caractériser l'effet des droits exogènes sur l'agrégation des préférences individuelles.

2.3.1 Extension des propriétés basiques classiques

La propriété classique fondamentale dont nous avons parlé au chapitre 1 est la propriété d'unanimité, ou en d'autres termes de compatibilité de l'ordre social avec la relation de Pareto. Cette propriété n'est pas influencée par l'introduction des droits exogènes inégaux, et donc son expression reste la même.

L'autre propriété classique dont nous avons parlé, et qui est très souvent requise surtout dans lorsque l'équité est au centre du problème de décision collective, est la propriété d'anonymat. Cette propriété traduit le principe informel selon lequel la qualité d'une décision collective est insensible à l'identité des agents, et s'exprime, dans le cas classique, par la symétrie de l'ordre de bien-être collectif. Cependant, lorsque l'on introduit des droits exogènes inégaux, l'identité d'un agent est révélée non seulement par la place des utilités dans le profil d'utilité, mais aussi par la place des droits dans le vecteur de droits inégaux. La propriété d'anonymat est donc redéfinie comme suit :

Définition 2.5 (Anonymat généralisé) *Soit $\succeq_{\overrightarrow{e}}$ un ordre de bien-être collectif à droits inégaux. $\succeq_{\overrightarrow{e}}$ satisfait la propriété d'*anonymat généralisé *si et seulement si $\forall(\overrightarrow{u}, \overrightarrow{v}) \in \mathscr{V}^n \times \mathscr{V}^n$, σ permutation de $\mathscr{N}$, et $(\overrightarrow{u}', \overrightarrow{v}') \in \mathscr{V}^n \times \mathscr{V}^n$ tels que $\overrightarrow{u}' = (u_{\sigma(1)}, \ldots, u_{\sigma(n)})$ et $\overrightarrow{v}' = (v_{\sigma(1)}, \ldots, v_{\sigma(n)})$, on a $\overrightarrow{u} \succeq_{\overrightarrow{e}} \overrightarrow{v} \Leftrightarrow \overrightarrow{u}' \succeq_{(e_{\sigma(1)},\ldots,e_{\sigma(n)})} \overrightarrow{v}'$.*

La propriété suivante concerne l'indépendance de l'utilité collective vis-à-vis des agents non concernés par une décision (IUA). Rappelons que cette propriété exprime le fait qu'un agent n'étant pas concerné par le choix entre deux décisions — parce que son utilité reste la même entre les deux décisions — ne doit pas être pris en compte pour ce choix. Nous proposons une propriété plus forte dans le cadre des droits inégaux : non seulement l'utilité de l'agent n'influence pas le choix entre les deux décisions, mais son droit exogène non plus.

Définition 2.6 (IUA généralisée) *Un ordre de bien être collectif à droits inégaux $\succeq_{\overrightarrow{e}}$ satisfait la propriété d'*indépendance généralisée vis-à-vis des agents non concernés *si et seulement si pour tout quadruplet de profils d'utilité $(\overrightarrow{u}, \overrightarrow{v}, \overrightarrow{u}', \overrightarrow{v}')$ et toute paire de vecteurs de droits $(\overrightarrow{e}, \overrightarrow{e}')$ tels que :*

▷ *pour un agent i : $u_i = v_i$ et $u'_i = v'_i$,*

▷ *pour tout agent $k \neq i$: $u_k = u'_k$, $v_k = v'_k$, et $e_k = e'_k$,*

nous avons :

$$\overrightarrow{u} \preceq_{\overrightarrow{e}} \overrightarrow{v} \Leftrightarrow \overrightarrow{u}' \preceq_{\overrightarrow{e}'} \overrightarrow{v}'.$$

$\overrightarrow{e}'$, $\overrightarrow{u}'$ et $\overrightarrow{v}'$ sont des répliques de $\overrightarrow{e}$, $\overrightarrow{u}$ et $\overrightarrow{v}$ sauf pour l'agent i. Entre $\overrightarrow{u}$ et $\overrightarrow{v}$, l'agent i n'est pas concerné car son utilité ne change pas ($u_i = v_i$). De même, cet agent n'est pas concerné par la décision entre $\overrightarrow{u}'$ et $\overrightarrow{v}'$, pour la même raison ($u'_i = v'_i$), *même si son droit a pu changer.* Dans ces conditions, si collectivement on préfère $\overrightarrow{u}$ à $\overrightarrow{v}$ sous $\overrightarrow{e}$, alors on doit préférer $\overrightarrow{u}'$ à $\overrightarrow{v}'$ sous $\overrightarrow{e}'$, si la propriété est respectée.

Si cette propriété n'est pas vérifiée, alors le choix entre deux décisions dépendra de l'utilité ou du droit de l'agent i, même si cet agent est complètement indifférent entre ces deux décisions, ce qui intuitivement est non souhaitable.

2.3.2 Équité en présence de droits exogènes inégaux

L'extension des propriétés classiques d'unanimité, d'anonymat et d'indépendance vis-à-vis des agents non concernés introduites précédemment était relativement évidente. Intéressons-nous maintenant aux propriétés relatives à l'équité introduites dans leur version classique dans le chapitre 1. L'extension de ces propriétés est légèrement plus délicate, car la notion d'équité est d'autant plus difficile à formaliser qu'elle est maintenant biaisée par l'introduction de l'inégalité des droits exogènes.

2.3.2.1 Juste part et absence d'envie

La propriété de juste part garantie est celle dont l'extension est la plus immédiate. Dans le cas classique, cette propriété s'appuie sur ce que chaque agent juge comme étant la part qui lui revient de droit, c'est-à-dire qu'il s'attend à obtenir le $n^{\text{ème}}$ de la valeur à laquelle il estime la ressource en entier. En présence d'une inégalité de droits exogènes, tous les agents ne peuvent pas prétendre à la même part de la ressource : la juste part des agents doit être proportionnelle à leur droit. En d'autres termes, la juste part d'un agent i correspond à la fraction $\overline{\underline{e_i}}$ de la valeur à laquelle il estime l'intégralité de la ressource. Si nous prenons l'exemple d'un partage impliquant deux agents dont les droits respectifs sont 1 et 3, leurs justes parts estimées respectives seront $\widehat{u_1}/4$ et $3\widehat{u_2}/4$.

Définition 2.7 (Test de juste part généralisé) *Soient $\mathscr{N}$ un ensemble d'agents, $\overrightarrow{e}$ un vecteur de droits exogènes, $\mathscr{A}$ l'ensemble des partages admissibles et $\overrightarrow{\pi}$ un partage. $\overrightarrow{\pi}$ satisfait le* test de juste part généralisé *si et seulement si $\forall i,\ f_i(\pi_i) \geq \overline{\underline{e_i}}\widehat{u_i}$, avec $\widehat{u_i} \stackrel{def}{=} \max\{f_i(\pi_i) | \overrightarrow{\pi} \in \mathscr{A}\}$.*

Cette propriété de juste part en présence de droits exogènes inégaux découle directement du principe de duplication des agents, appliqué au cas où la fonction de répartition est la division : un partage concernant n agents satisfait le test de juste part avec droits exogènes si et seulement si le partage correspondant aux m clones satisfait le test de juste part classique. Chacun des clones d'un agent i estimant qu'il a au moins le $m^{\text{ème}}$ de la ressource selon son propre point de vue (qui est aussi celui de l'agent i initial), l'agent i estimera qu'il a une fraction de la ressource au moins égale à e_i/m. Cette propriété correspond au principe de proportionnalité introduit pour des droits inégaux dans [Steinhaus, 1948], cité et repris de manière implicite dans [Brams et Taylor, 1996].

Bien entendu, on peut de la même manière étendre la propriété de juste part garantie introduite au chapitre 1 :

Définition 2.8 (Juste part garantie généralisée) *Soient $\mathscr{N}$ un ensemble d'agents, $\overrightarrow{e}$ un vecteur de droits exogènes, $\mathscr{A}$ l'ensemble des partages admissibles, $\succeq_{\overrightarrow{e}}$ un ordre de bien-être collectif à droits inégaux et $\widehat{\mathscr{A}}$ l'ensemble des solutions non dominées pour cet ordre collectif. Soit $\mathscr{F}_{\overrightarrow{e}} = \{\overrightarrow{\pi} \in \mathscr{A} | \forall i \in [\![1, n]\!],\ f_i(\pi_i) \geq \overline{\underline{e_i}}\widehat{u_i}\}$. $\succeq_{\overrightarrow{e}}$ vérifie la propriété de* juste part garantie généralisée *si et seulement si $\mathscr{F}_{\overrightarrow{e}} \neq \varnothing \Rightarrow \widehat{\mathscr{A}} \cap \mathscr{F} \neq \varnothing$.*

Intéressons-nous maintenant à la propriété d'absence d'envie, correspondant dans le cas classique à une certaine vision de l'équité fondée sur la comparaison personnelle (interne à chaque agent) de la propre part d'un agent et de la part des autres agents. À première vue, ce principe est incompatible avec l'idée même d'inégalité des droits exogènes. En effet, si un agent i a un droit beaucoup plus élevé qu'un autre agent j, il ne semble pas anormal qu'il ait une part beaucoup plus importante de la ressource que l'agent j, suscitant ainsi à coup sûr l'envie de cet agent. C'est le sens de la remarque de [Young, 1994] à propos de l'absence d'envie, citée par [Brams et Taylor, 1996] : « [envy-freeness] only applies when the parties have equal claims on the good ».

Cependant, une autre vision de l'absence d'envie est possible dans ce contexte, comme le fait remarquer [Brams et Taylor, 1996], vision qui est encore une fois inspirée du principe de duplication des agents avec fonction de répartition égale à la division. De manière informelle, un partage concernant n agents est sans envie si et seulement si aucun des clones n'envie la part d'un autre clone. Cette propriété a été formalisée de la manière suivante dans [Pivato, 2006] :

Définition 2.9 (Test d'absence d'envie généralisé) *Soit $\mathscr{N}$ un ensemble d'agents, $\overrightarrow{e}$ un vecteur de droits exogènes, et $\{f_1, \ldots, f_n\}$ l'ensemble de leurs fonctions d'utilités exprimées sur les* *parts et $\overrightarrow{\pi}$ un partage. $\overrightarrow{\pi}$ satisfait le* test d'absence d'envie généralisé *si et seulement si* $\forall i \neq j$ *:*

$$\frac{f_i(\pi_i)}{e_i} \geq \frac{f_i(\pi_j)}{e_j}.$$

Un tel partage sera dit $\overrightarrow{e}$-sans envie.

Autrement dit, tout agent i s'estime plus satisfait relativement à son droit e_i qu'il ne le serait avec la part d'un autre agent j, relativement au droit e_j de cet autre agent.

2.3.2.2 Équité fondée sur l'égalitarisme et droits inégaux

Réduction des inégalités et courbe de Lorenz Si les deux notions précédentes (juste part et absence d'envie) ont déjà été introduites dans la littérature sur le partage, en revanche la plupart des travaux traitant de la mesure des inégalités dans le domaine de la décision collective se limitent au cas classique, c'est-à-dire au cas où les droits exogènes sont égaux. On trouve cependant une exception dans la littérature économique : l'article [Ebert et Moyes, 2002], qui introduit une extension des travaux sur les mesures d'inégalités permettant de prendre en compte une pondération sur les agents[4]. Il est très intéressant de remarquer que ce travail se situe dans un contexte très différent du nôtre, même s'il y a des similarités formelles claires. Alors que l'objectif de notre travail se concentre sur la prise en compte de droits inégaux dans la procédure de partage, le travail de [Ebert et Moyes, 2002] a pour objet la mesure des inégalités *a posteriori* sur un ensemble de revenus relevés dans une population. Les poids ont d'autres significations dans ce contexte : ils sont introduits par exemple pour pallier les faiblesses de la procédure de mesure qui fournit le profil de revenus (échantillonnage non représentatif de la population par exemple, sous-représentativité d'une certaine classe d'individus, etc.). En d'autres termes, alors que dans notre cas les droits exogènes ont un rôle d'*équité*, dans le travail cité les poids ont un rôle *correctif* de l'imperfection de l'information relevée.

Les différences que nous avons mises en valeur entre l'approche de [Ebert et Moyes, 2002] et la nôtre ont deux conséquences. Tout d'abord, rien n'indique, dans le travail de [Ebert et Moyes, 2002], que les profils d'utilité à comparer soient de taille fixe, ni que les poids soient fixés à l'avance. On peut avoir par exemple à comparer deux populations sur la donnée de deux profils de revenus de taille et de vecteurs de poids différents. Autre différence d'approche, chaque «utilité» des profils à comparer dans le travail de [Ebert et Moyes, 2002] correspond au revenu d'un individu d'une certaine classe de personnes : selon notre point de vue de la duplication des agents, ce n'est pas une somme d'argent à partager mais cela représente l'«agrément» d'être dans une classe de la population (la «prospérité», si nous reprenons les termes de la section 2.1.2). Transposé dans notre cadre, le travail de [Ebert et Moyes, 2002] correspond donc à une vision «réplication» de la fonction de répartition, alors que notre approche est fondée sur la fonction «division». Le parallèle est néanmoins intéressant : même si la différence d'approche conduit à des définitions différentes, la démarche est similaire, et les définitions sont équivalentes «à la fonction de répartition près».

Comme pour la juste part et l'absence d'envie, nous allons donc nous inspirer du principe de duplication des agents avec fonction de réplication égale à la division afin de nous ramener au cas classique. La justification est qu'en présence de droits exogènes inégaux, le lieu des profils parfaitement égalitaires correspond à la droite $u_1/e_1 = \cdots = u_n/e_n$, c'est-à-dire aux profils parfaitement égalitaires au sens classique du terme pour les m clones.

Nous inspirant de ce principe, nous pouvons étendre la notion de transfert de Pigou-Dalton en y introduisant les droits exogènes inégaux comme suit :

[4] Nous avons découvert cet article après notre travail sur la formalisation des droits exogènes inégaux ; le parallèle entre cette approche et la nôtre est tout-à-fait intéressant.

Définition 2.10 (Transfert de Pigou-Dalton (réduction des inégalités)) *Soient $\overrightarrow{u}$ et $\overrightarrow{u}'$ deux profils d'utilité et $\overrightarrow{e}$ un vecteur de droits exogènes. $\overrightarrow{u}'$ est obtenu à partir de $\overrightarrow{u}$ par* réduction des inégalités généralisée *relativement à $\overrightarrow{e}$, ou* transfert de Pigou-Dalton généralisé *si et seulement si $\exists(i,j) \in \mathscr{N}^2$ tels que :*
- ▷ $i \neq j$;
- ▷ $\overrightarrow{u_i} + \overrightarrow{u_j} = \overrightarrow{u_i}' + \overrightarrow{u_j}'$ *(conservation de la somme) ;*
- ▷ $\frac{u_i}{e_i} < \{\frac{u'_i}{e_i}, \frac{u'_j}{e_j}\} < \frac{u_j}{e_j}$ *(réduction des inégalités) ;*
- ▷ $\forall k \in \mathscr{N} \setminus \{i,j\}$, $u_k = u'_k$.

La notion de principe de réduction des inégalités généralisé suit naturellement :

Définition 2.11 (Principe de réduction des inégalités généralisé) *Soit $\succeq_{\overrightarrow{e}}$ un ordre de bien-être collectif à droits exogènes. $\succeq_{\overrightarrow{e}}$ satisfait le* principe de réduction des inégalités généralisé *si et seulement si pour toute paire de profils d'utilité $\overrightarrow{u}$, $\overrightarrow{u}'$ tels que $\overrightarrow{u}'$ est obtenu à partir de $\overrightarrow{u}$ par transfert de Pigou-Dalton relativement à $\overrightarrow{e}$, on a $\overrightarrow{u} \prec_{\overrightarrow{e}} \overrightarrow{u'}$.*

Voici comment nous interprétons ce principe en présence de droits exogènes inégaux. Dans ce contexte, un transfert de Pigou-Dalton est simplement un transfert réduisant les inégalités entre les clones, avec la contrainte que l'utilité des clones correspondant au même agent initial doit rester la même. Une réduction des inégalités entre les clones de l'agent i et de l'agent j correspond donc à un transfert d'utilité tel que chaque clone de l'agent j donne une utilité de ε/e_j, et chaque clone de l'agent i reçoit une utilité de ε/e_i. Le transfert total d'utilité, correspondant à ε, se fait donc à somme $u_i + u_j$ constante.

Bien entendu, à l'échelle du problème impliquant les m clones, un tel transfert n'est pas un transfert de Pigou-Dalton au sens strict du terme, car en particulier il n'implique pas uniquement deux agents. Cependant, on peut facilement vérifier que tout transfert de ce type peut être obtenu par une séquence de transferts de Pigou-Dalton. Considérons par exemple le cas d'un transfert de Pigou-Dalton avec droits exogènes d'un agent i ayant un droit de 3 vers un agent j ayant un droit de 2. Supposons que l'agent i transfère une utilité ε vers l'agent j. À l'échelle des clones, cela revient à un transfert de $\varepsilon/3$ de chaque clone de l'agent i vers les clones de l'agent j. Ce transfert peut s'effectuer de la manière suivante :
- ▷ Le clone i_1 transfère $\varepsilon/3$ vers le clone j_1.
- ▷ Le clone i_2 transfère $\varepsilon/3$ vers le clone j_2.
- ▷ Le clone i_3 transfère $\varepsilon/6$ vers le clone j_1.
- ▷ Le clone i_3 transfère $\varepsilon/6$ vers le clone j_2.

Au final, chacun de ces transferts est de type Pigou-Dalton, et le transfert total correspond bien à une réduction des inégalités au sens des droits exogènes inégaux.

Nous pouvons nous appuyer sur cette extension du principe de réduction des inégalités pour définir la notion de courbe de Lorenz avec droits inégaux :

Définition 2.12 (Courbe de Lorenz avec droits inégaux) *Soit $\overrightarrow{u}$ un vecteur d'utilités et $\overrightarrow{e}$ un vecteur de droits inégaux. La* courbe de Lorenz avec droits inégaux *de $\overrightarrow{u}$ est le vecteur :*

$$\overrightarrow{L_{\overrightarrow{e}}(\overrightarrow{u})} = \overrightarrow{L(\overrightarrow{u} \triangleright^{div}_{\overrightarrow{e}})}, \text{ avec } \overrightarrow{u} \triangleright^{div}_{\overrightarrow{e}} \stackrel{def}{=} \big(\underbrace{u_1/e_1, \ldots, u_1/e_1}_{e_1 \text{ fois}}, \ldots, \underbrace{u_n/e_n, \ldots, u_n/e_n}_{e_n \text{ fois}}\big).$$

Si $\overrightarrow{L_{\overrightarrow{e}}(\overrightarrow{u})}$ domine $\overrightarrow{L_{\overrightarrow{e}}(\overrightarrow{u})}$ au sens de Pareto, on dira que $\overrightarrow{u}$ $\overrightarrow{e}$-Lorenz-domine $\overrightarrow{v}$.

Bien entendu, étant donnée la construction de la notion de courbe de Lorenz à droits inégaux, nous avons toujours l'équivalence entre le fait que $\overrightarrow{u}$ $\overrightarrow{e}$-Lorenz-domine $\overrightarrow{v}$ et le fait que $\overrightarrow{u}$ puisse être obtenue à partir de $\overrightarrow{v}$ par une séquence de transferts de Pigou avec droits exogènes et d'améliorations de Pareto.

Le principe de fonctionnement de la dominance de Lorenz en présence de droits exogènes inégaux est illustré sur la figure 2.2. Nous pouvons vérifier d'une part que la droite de parfaite égalité n'est plus la droite $u_1 = u_2$, mais bien la droite $u_1/e_1 = u_2/e_2$, et d'autre part qu'il y a une asymétrie de la zone de réduction des inégalités, en faveur de l'agent ayant le droit le plus élevé : en l'occurrence u_1. L'introduction de droits inégaux dans le principe de réduction des inégalités ne se résume donc pas à une simple dilatation des échelles individuelles d'utilités inversement proportionnelle aux droits inégaux.

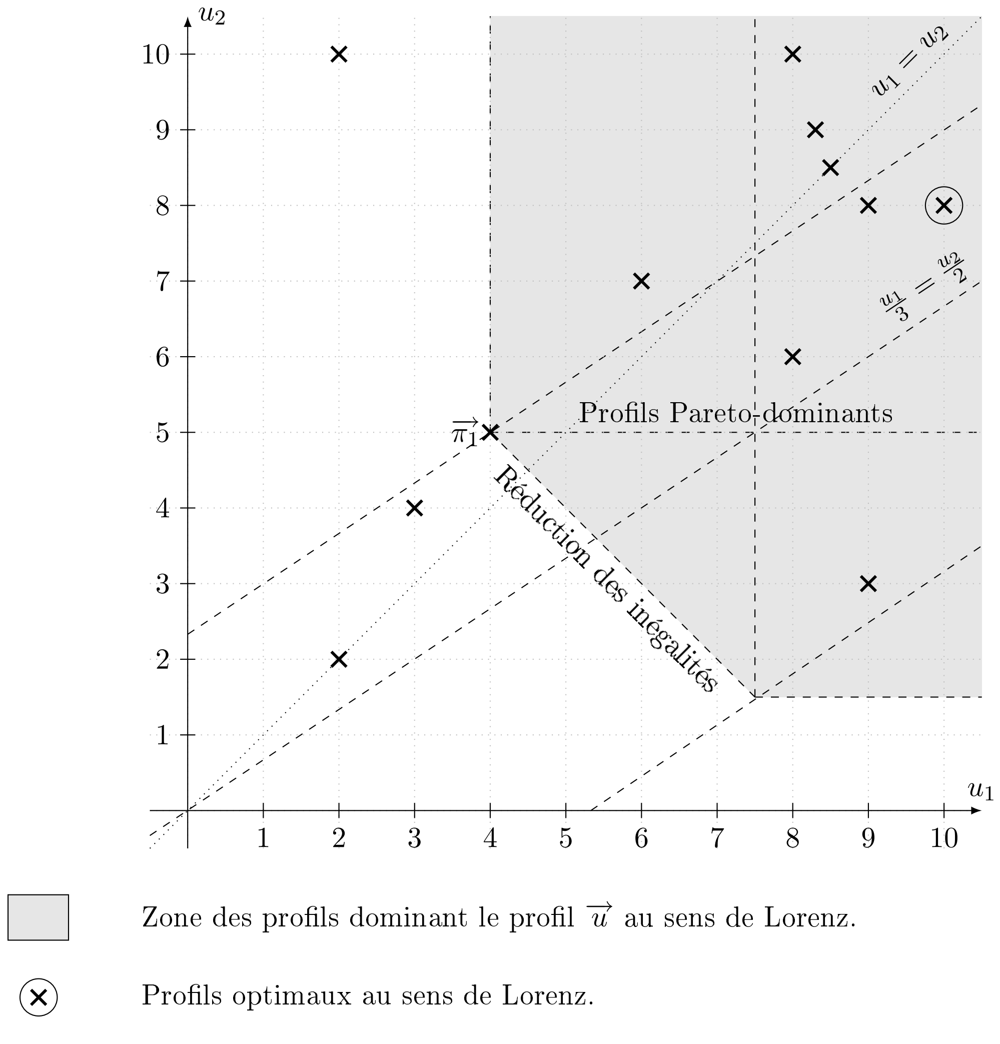


Figure 2.2 — Illustration de la notion de dominance de Lorenz avec droits exogènes inégaux sur des profils d'utilité à deux composantes. Le vecteur de droits exogènes est $(3, 2)$.

Indices d'inégalité Les bases liées à la définition des inégalités étant posées, l'extension de la définition des indices d'inégalité ne pose pas réellement problème :

Définition 2.13 (Indice d'inégalité avec droits exogènes) *Soit $\succeq_{\overrightarrow{e}}$ un ordre de bien-être collectif à droits inégaux qui respecte le principe de réduction des inégalités avec droits exogènes. Pour chaque vecteur d'utilité positif $\overrightarrow{u}$ on définit l'utilité également distribuée équivalente $\varepsilon_{\overrightarrow{e}}(\overrightarrow{u}) \in \mathscr{V}$ de la manière suivante : $\varepsilon_{\overrightarrow{e}}(\overrightarrow{u})\cdot(e_1,\ldots,e_n) \sim_{\overrightarrow{e}} \overrightarrow{u}$. On note également $\overline{u\triangleright^{div}_{\overrightarrow{e}}} = \frac{1}{m}\sum_{i=1}^{n} u_i$. L'$\overrightarrow{e}$*-indice d'inégalité *associé à $\succeq_{\overrightarrow{e}}$ est :*

$$J_{\overrightarrow{e}}(\overrightarrow{u}) = 1 - \frac{\varepsilon_{\overrightarrow{e}}(\overrightarrow{u})}{\overline{u\triangleright^{div}_{\overrightarrow{e}}}}.$$

Comme dans le cas classique, le fait que l'ordre de bien-être collectif associé à l'indice d'inégalité satisfait le principe de réduction des inégalités assure que $J_{\overrightarrow{e}}(\overrightarrow{u}) \geq 0$. Cet indice d'inégalité est nul si et seulement si les composantes de $\overrightarrow{u}\triangleright^{div}_{\overrightarrow{e}}$ sont toutes égales, c'est-à-dire si le vecteur $\overrightarrow{u}$ est colinéaire au vecteur de droits $\overrightarrow{e}$.

L'extension de la définition de l'indice d'Atkinson se fait grâce au même principe (le détail des calculs est reporté en annexe C).

$$\begin{cases} J_{q,\overrightarrow{e}}(\overrightarrow{u}) = 1 - \dfrac{m}{n}\left(\dfrac{1}{m}\displaystyle\sum_{i=1}^{n} e_i^{1-q}\left(\dfrac{u_i}{\overline{u}}\right)^q\right)^{\frac{1}{q}}, & 0 < q < 1 \text{ ou } q < 0 \\ J_0(\overrightarrow{u}) = 1 - \dfrac{m}{n}\displaystyle\prod_{i=1}^{n}\left(\dfrac{u_i}{e_i\overline{u}}\right)^{\overline{e_i}}. \end{cases}$$

Ces indices d'inégalité sont fondés sur la généralisation de la famille somme des puissances aux droits exogènes inégaux, que nous allons introduire plus loin.

De même, pour l'indice de Gini :

$$\begin{aligned} G_{\overrightarrow{e}}(\overrightarrow{u}) &= \frac{1}{2nm\overline{u}} \sum_{1\leq k,l\leq m} e_k e_l \left|\frac{u_k}{e_k} - \frac{u_l}{e_l}\right| \\ &= 1 - \frac{1}{nm\overline{u}}\sum_{k=1}^{n}(2m+1-(2E_{k-1}+e_k^{\uparrow\frac{u}{e}})(e_k^{\uparrow\frac{u}{e}}+1))\left(\frac{u_k}{e_k}\right)^{\uparrow\frac{u}{e}}, \end{aligned}$$

avec $\overrightarrow{x}^{\uparrow\frac{u}{e}}$ désignant le vecteur des composantes de $\overrightarrow{x}$ réordonnées selon l'ordre des composantes de $\overrightarrow{\frac{u}{e}}$, et $x_k^{\uparrow\frac{u}{e}}$ désignant la $k^{\text{ème}}$ composante de ce vecteur trié.

L'interprétation de ces trois formes équivalentes de l'indice de Gini est relativement similaire au cas classique. La première forme définit l'aire normalisée entre la courbe de Lorenz généralisée réelle et la courbe de Lorenz généralisée idéale, correspondant à un profil parfaitement égalitaire à l'échelle des clones. La deuxième forme correspond à la somme normalisée des distances entre les utilités des clones, et enfin la troisième forme correspond à une moyenne pondérée ordonnée à l'échelle des clones.

Réduction des inégalités et dominance stochastique Si la notion de réduction des inégalités n'a pas été réellement formalisée à notre connaissance dans le domaine de la décision collective, on peut en revanche effectuer un rapprochement avec la notion de dominance stochastique appliquée à des problèmes de décision entre présence de risque. Nous allons essayer d'introduire de manière très rapide les principales notions de ce domaine, et mettre en valeur le parallèle existant entre la dominance stochastique et la réduction des inégalités en présence de droits exogènes inégaux. L'objectif n'est pas de faire une description exhaustive et détaillée du domaine de la décision en présence de risque (on pourra se référer à des ouvrages tels que [Bouyssou *et al.*, 2006; Gayant, 2001]), mais d'exploiter le parallèle entre ce domaine et celui de la décision collective pour donner un autre éclairage sur la notion d'inégalité en présence de droits exogènes inégaux.

La décision en présence de risque est le sous-domaine de la décision dans lequel le choix se fait entre des alternatives à l'issue incertaine, mais dont la probabilité d'occurrence est connue avec précision[5]. La théorie sous-jacente est donc ici la théorie des probabilités. Nous rappelons quelques définitions essentielles.

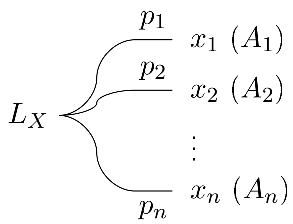
Définition 2.14 (Variable aléatoire discrète) *Soit Ω un espace d'états de la nature fini, et $\mathscr{I}$ un espace d'issues (outcome en Anglais), par exemple $\mathbb{R}$. Une* variable aléatoire discrète *est une fonction $X : \Omega \to \mathscr{I}$.*

Puisque Ω est fini, il existe une partition $(A_1, \dots, A_n)$ de Ω telle que $\forall i, X(\omega) = x_i$ pour tout $\omega \in A_i$. On note $X = (x_1, A_1; \dots ; x_n, A_n)$.

Lorsque l'on connaît avec précision les probabilités d'occurrence de tous les évènements A_i, on peut assimiler la variable aléatoire X à sa distribution de probabilité, appelée loterie :

Définition 2.15 (loterie) *Étant donnée une variable aléatoire $X = (x_1, A_1; \dots ; x_n, A_n)$ sur un espace Ω, et une mesure de probabilité P sur Ω, on appellera* loterie *associée à X sa distribution de probabilité. Elle sera notée $(x_1, p_1; \dots ; x_n, p_n)$, avec $p_i = P(A_i)$ (et $\sum_{i=1}^{n} p_i = 1$). Nous noterons $\mathcal{L}$ l'ensemble des loteries sur Ω.*

On représente traditionnellement une loterie sous la forme d'un arbre constitué des feuilles correspondant à chaque évènement possible, associé à sa probabilité.



Pour une variable aléatoire discrète, on peut définir les fonctions cumulative et décumulative de probabilité :

Définition 2.16 *Étant donnée une variable aléatoire $X = (x_1, A_1; \dots ; x_n, A_n)$ sur un espace Ω, et une mesure de probabilité P sur Ω, la* fonction cumulative de probabilité *de X, notée F_X, est l'application de $\mathscr{I}$ dans $[0; 1]$ telle que $F_X : x \mapsto P(\omega \in \Omega | X(\omega) \leq x)$. La* fonction décumulative de probabilité *de X est notée G_X et est définie comme suit : $G_X = 1 - F_X$.*

Le modèle à la base du domaine de la prise de décision en présence de risque est fondé sur la notion de relation de préférence d'un agent sur les loteries. Les préférences d'un agent sont simplement représentées par un préordre total sur l'ensemble des loteries $\mathcal{L}$, ou une fonction d'utilité $U : \mathcal{L} \to \mathscr{V}$.

Le rapprochement formel entre la décision en présence de risque et la décision collective est connu. Il est fondé sur le parallèle exprimé dans le tableau 2.1 (nous supposons que les probabilités des loteries sont exprimées sous forme de nombres rationnels).

[5]Contrairement au domaine de la décision dans l'incertain, qui concerne les problèmes de décision à l'issue incertaine mais dont on ne sait pas quantifier les probabilités d'occurrence d'évènements.

Issue	Agent
Décision en présence de risque	Décision collective
État de la nature	Décision collective
Loterie	Profil d'utilités
Utilité sur les issues	Utilité individuelle
Utilité sur les loteries	Utilité collective
Probabilité	Pourcentage de population ou droit exogène

Tableau 2.1 — Rapprochement formel entre la décision en présence de risque et la décision collective.

Ainsi, par exemple, une loterie $(10, 1/4; 20, 1/4; 50, 1/2)$ pourra se traduire en terme de décision collective par une décision dans laquelle 1/4 de la population reçoit 10, 1/4 de la population reçoit 20 et la moitié de la population reçoit 50. Ce parallèle va nous permettre d'illustrer la notion d'inégalités dans la décision collective à l'aide des notions de dominance stochastique, issues du domaine des probabilités.

Définition 2.17 (Dominance stochastique du premier ordre) *Soient deux variables aléatoires quelconques définies sur un même ensemble d'états de la nature Ω. Alors X domine la variable Y au sens de la dominance stochastique du premier ordre (que l'on notera $X \succeq_{FOSD} Y$) si et seulement si :*

$$\forall x \in \mathscr{I}, \; P(X > x) \geq P(Y > x).$$

En d'autres termes : $\forall x \in \mathscr{I}$, $G_X(x) \geq G_Y(x)$.

Cela signifie qu'en toute circonstance X est meilleur que Y, c'est-à-dire que pour tout x, la probabilité de gagner au moins x est plus forte avec X qu'avec Y. Transposé au contexte de la décision collective, cela se traduit par le fait que pour tout seuil d'utilité u, le pourcentage de la population (ou le nombre de clones) ayant une utilité supérieure ou égale à u est plus élevé pour $\overrightarrow{\pi}$ que pour $\overrightarrow{\pi}'$.

La relation de dominance stochastique qui nous intéresse plus dans le contexte de la mesure des inégalités est la dominance stochastique du second ordre :

Définition 2.18 (Dominance stochastique du second ordre) *Soient deux variables aléatoires quelconques définies sur un même ensemble d'états de la nature Ω. Alors X domine la variable Y au sens de la dominance stochastique du second ordre (que l'on notera $X \succeq_{SOSD} Y$) si et seulement si :*

$$\forall x \in \mathscr{I}, \; \int_x^{+\infty} P(X > t)dt \geq \int_x^{+\infty} P(Y > t)dt.$$

En d'autres termes : $\forall x \in \mathscr{I}$, $\int_x^{+\infty} G_X(t)dt \geq \int_x^{+\infty} G_Y(t)dt$.

En gros, cela correspond à la notion d'étalement de la distribution de probabilité : plus une distribution est étalée, moins elle est préférée. Remarquons que ce concept n'est pas tout-à-fait identique à la notion d'écart-type : par exemple, les deux distributions de probabilité $(8, 3/4; 12, 1/4)$ et $(10, 1/4; 14, 3/4)$ ont le même écart-type, mais la deuxième distribution domine la première au sens stochastique du second ordre.

Dans le domaine de la décision en présence de risque, on traduit généralement la notion d'*aversion pour le risque* d'un agent à l'aide de cette relation de dominance stochastique. Plus précisément, l'aversion au risque est traduite par la notion d'*étalement à moyenne constante*, fondée sur la dominance stochastique du second ordre :

Définition 2.19 (Étalement à moyenne constante) *Étant données deux variables aléatoires X et Y dont les espérances sont respectivement $E(X)$ et $E(Y)$, on dira que Y se déduit de X par un* étalement à moyenne constante *(noté $X \rightsquigarrow_{MPS} Y$) si et seulement si :*

1. $E(X) = E(Y)$, *et*
2. $X \succeq_{SOSD} Y$.

Cette notion traduit une aversion forte pour un accroissement du risque. Dans la définition précédente, la distribution de probabilité de Y est plus «étalée» que celle de X, à espérance identique. Autrement dit, dans une telle loterie, un agent risque de gagner plus, mais il risque aussi de perdre plus. Cette notion est illustrée de manière graphique sur la figure 2.3.2.2.

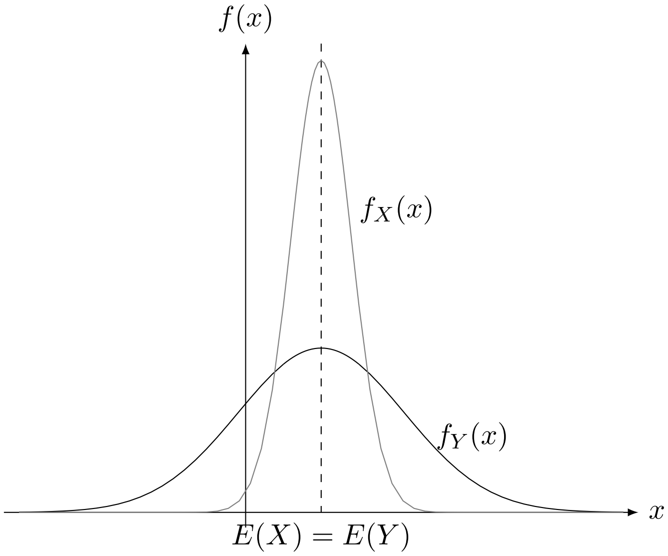


Figure 2.3 — Deux exemples de distributions de probabilité (continues) de fonctions de densité f_X et f_Y. Y est obtenu à partir de X par un étalement à moyenne constante.

Il est encore une fois possible de se ramener au domaine de la décision collective. Cette notion d'aversion au risque correspond formellement à la notion d'aversion à l'inéquité. Un étalement à moyenne constante correspond, dans le domaine de la décision collective, à un transfert d'utilité entre les clones, qui a pour effet d'«étaler» le profil d'utilité, c'est-à-dire d'accroître les inégalités entre les agents. Un tel transfert se fait bien entendu à somme des utilités constante, car l'étalement à moyenne constante se fait à espérance mathématique constante. Il s'agit donc de l'inverse d'un transfert de Pigou-Dalton généralisé (autrement dit, c'est un transfert qui accroît les inégalités).

Ce parallèle nous permet de valider notre approche de l'inégalité en présence de droits inégaux, en la rapprochant de la notion de risque avec des probabilités inégales sur les évènements : cette notion de risque est définie à l'aide du concept d'étalement à moyenne constante, qui correspond de manière formelle à ce que nous avons défini comme étant (l'inverse d') un transfert de Pigou-Dalton généralisé. L'accroissement des inégalités en présence de droits exogènes inégaux correspond donc formellement au concept d'accroissement des risques en présence de probabilités inégales.

2.3.3 Nouvelles propriétés relatives aux droits exogènes

Le concept de droit exogène exprime la simple notion d'influence plus ou moins grande d'un agent sur la décision collective, avec l'idée intuitive qu'un agent doté d'un droit plus grand doit tirer un bénéfice (donc une utilité) plus grand de la décision collective. Cette idée informelle peut se traduire de différentes manières. La propriété la plus simple et la plus évidente requise par l'introduction de droits inégaux est que l'augmentation du droit d'un agent ne peut pas renverser une préférence collective qui déjà l'avantageait.

Définition 2.20 (Conformité) *Soient $\overrightarrow{e}$ et $\overrightarrow{e}'$ deux vecteurs de droits tels que $e_k = e'_k$ pour tout $k \neq i$, et $e_i < e'_i$, et soit $\succeq_{\overrightarrow{e}}$ un ordre de bien-être collectif à droits inégaux. Alors $\succeq_{\overrightarrow{e}}$ vérifie la propriété de* conformité *si et seulement si pour toute paire de profils d'utilité $(\overrightarrow{u}, \overrightarrow{v})$, on a $(\overrightarrow{u} \succeq_{\overrightarrow{e}} \overrightarrow{v}$ et $u_i > v_i) \Rightarrow \overrightarrow{u} \succeq_{\overrightarrow{e}'} \overrightarrow{v}$.*

Cette définition est plus claire sur un exemple :

Exemple 2.1 Soient $\overrightarrow{u} = (4, 7, 4, 2)$ et $\overrightarrow{v} = (1, 5, 3, 8)$. Supposons que, pour un vecteur de droits $\overrightarrow{e}$, on ait $\overrightarrow{u} \succeq_{\overrightarrow{e}} \overrightarrow{v}$. Entre les deux vecteurs, le préféré est celui qui avantage, entre autres, l'agent 2. Maintenant si nous augmentons le droit de l'agent 2 sans modifier celui des autres agents, ce qui nous donne le vecteur $\overrightarrow{e}'$, nous ne pouvons avoir $\overrightarrow{v} \succeq_{\overrightarrow{e}} \overrightarrow{u}$, car si tel était le cas, la collectivité préférerait désormais un profil d'utilité qui désavantage maintenant l'agent 2, alors que son droit a augmenté.

L'implication principale de cette propriété concerne le calcul d'une décision collective optimale :

Proposition 2.1 *Soient $\overrightarrow{e}$ et $\overrightarrow{e}'$ deux vecteurs de droits tels que $e_k = e'_k$ pour tout $k \neq i$, et $e_i < e'_i$, et soit $\succeq_{\overrightarrow{e}}$ un ordre de bien-être collectif à droits inégaux. Nous notons $\widehat{\overrightarrow{\pi}}_{\overrightarrow{e}} \in \mathrm{argmax}^{\succeq_{\overrightarrow{e}}}_{\overrightarrow{\pi} \in \mathscr{A}}(\overrightarrow{f}(\overrightarrow{\pi}))$ une décision collective optimale selon $\succeq_{\overrightarrow{e}}$. Si $\succeq_{\overrightarrow{e}}$ satisfait la propriété de conformité, alors il existe une décision collective optimale $\widehat{\overrightarrow{\pi}}_{\overrightarrow{e}'}$ selon $\succeq_{\overrightarrow{e}'}$ telle que $f_i(\widehat{\overrightarrow{\pi}}_{\overrightarrow{e}'}) \geq f_i(\widehat{\overrightarrow{\pi}}_{\overrightarrow{e}})$.*

En d'autres termes, si l'on augmente le droit relatif d'un certain agent, alors il ne peut pas obtenir au final une utilité moindre qu'avant son augmentation.

> **Démonstration** Si $\max^{\succeq_{\overrightarrow{e}}}_{\overrightarrow{\pi} \in \mathscr{A}}(\overrightarrow{f}(\overrightarrow{\pi})) \sim_{\overrightarrow{e}'} \overrightarrow{f}(\hat{\pi}_{\overrightarrow{e}}))$, alors $\hat{\pi}_{\overrightarrow{e}}$ est aussi une décision optimale selon $\succeq_{\overrightarrow{e}'}$ et donc la proposition est satisfaite.
>
> Sinon, pour toute décision $\widehat{\overrightarrow{\pi}}_{\overrightarrow{e}'}$ optimale selon $\succeq_{\overrightarrow{e}'}$, on a $\overrightarrow{u}(\widehat{\overrightarrow{\pi}}_{\overrightarrow{e}'}) \succ_{\overrightarrow{e}'} \overrightarrow{u}(\widehat{\overrightarrow{\pi}}_{\overrightarrow{e}})$. Or, $\widehat{\overrightarrow{\pi}}_{\overrightarrow{e}}$ étant une décision optimale selon $\succeq_{\overrightarrow{e}}$, pour toute décision collective admissible $\overrightarrow{\pi}$ telle que $u_i(\overrightarrow{\pi}) \leq u_i(\widehat{\overrightarrow{\pi}}_{\overrightarrow{e}})$, on a $\overrightarrow{u}(\overrightarrow{\pi}) \preceq_{\overrightarrow{e}} \overrightarrow{u}(\widehat{\overrightarrow{\pi}}_{\overrightarrow{e}})$. D'après la propriété de conformité, on a donc aussi $\overrightarrow{u}(\overrightarrow{\pi}) \preceq_{\overrightarrow{e}'} \overrightarrow{u}(\widehat{\overrightarrow{\pi}}_{\overrightarrow{e}})$. Par contraposée, on a donc $u_i(\widehat{\overrightarrow{\pi}}_{\overrightarrow{e}}) \leq u_i(\widehat{\overrightarrow{\pi}}_{\overrightarrow{e}'})$, ce qui prouve la proposition. ▲

La propriété de conformité est une traduction possible de l'idée selon laquelle les droits inégaux ont un effet «positif» dans le partage : si l'on augmente le droit d'un agent, alors il ne pourra pas être désavantagé. Cette idée d'effet «positif» peut se traduire d'une manière différente : toutes choses étant égales par ailleurs, il vaut mieux choisir la décision qui avantage, entre deux agents ayant des droits inégaux, l'agent ayant un plus grand droit.

Définition 2.21 (Avantage aux droits élevés) *Soient $\overrightarrow{u}$ et $\overrightarrow{v}$ deux profils d'utilité tels que $u_i = v_j$, $u_j = v_i$ et $u_k = v_k$ $\forall k \in \mathscr{N} \setminus \{i, j\}$ ($\overrightarrow{v}$ est égal au profil $\overrightarrow{u}$ dans lequel on a permuté u_i et u_j), avec $u_i > u_j$, et soit $\succeq_{\overrightarrow{e}}$ un ordre de bien-être collectif à droits inégaux. Alors $\succeq_{\overrightarrow{e}}$* avantage les droits élevés *si et seulement si pour tout $\overrightarrow{e}$, $\overrightarrow{v} \preceq_{\overrightarrow{e}} \overrightarrow{u} \Leftrightarrow e_i \geq e_j$.*

Cette notion d'avantage aux droits élevés n'est pas équivalente à la propriété de conformité, même si elle exprime une traduction différente de la même idée intuitive. En effet, il existe des fonctions d'utilité collective à droits inégaux qui satisfont la propriété de conformité, sans toutefois vérifier la propriété d'avantage aux droits élevés. La fonction somme non pondérée, correspondant à l'utilitarisme avec division de la ressource, est un exemple d'une telle fonction. L'existence d'un lien d'implication entre l'avantage aux droits élevés et la conformité, éventuellement lié aux autres propriétés (IUA généralisée, anonymat, unanimité), ou aux propriétés analytiques des fonctions d'utilité collective (continuité, ...) n'est pas encore claire.

Une dernière propriété souhaitable des fonctions d'utilité collective prenant en compte des droits exogènes inégaux est leur insensibilité à une dilatation proportionnelle de l'échelle commune d'expression de ces droits inégaux. En d'autres termes, une fonction d'utilité collective à droits inégaux doit classer les décisions de la même manière, que le vecteur de droits soit $\overrightarrow{e}$, $2 \cdot \overrightarrow{e}$ ou bien $100 \cdot \overrightarrow{e}$.

Définition 2.22 (Insensibilité à une dilatation commune des droits (IDCD)) *Soit $\succeq_{\overrightarrow{e}}$ un ordre de bien-être collectif à droits inégaux. $\succeq_{\overrightarrow{e}}$ est* insensible à une dilatation commune des droits *(IDCD) si et seulement si $\forall k \in \mathbb{N}$, $\forall \overrightarrow{e}$ vecteur de droits, et pour tout couple $(\overrightarrow{u}, \overrightarrow{v})$ de profils d'utilité, $\overrightarrow{u} \succeq_{\overrightarrow{e}} \overrightarrow{v} \Leftrightarrow \overrightarrow{u} \succeq_{k \cdot \overrightarrow{e}} \overrightarrow{v}$.*

2.3.4 Application aux quatre ordres de bien-être social étendus

Nous nous intéressons maintenant à la caractérisation des quatre ordres de bien-être social étendus introduits en section 2.1.2 : l'ordre utilitariste et l'ordre égalitariste leximin, avec les fonctions de répartition division et réplication. La preuve de la proposition suivante est immédiate et ne présente que peu d'intérêt. Nous l'omettrons donc.

Proposition 2.2 *Les fonctions somme, somme pondérée,* leximin *et* leximin *pondéré satisfont les propriétés marquées «oui» dans la table 2.2, et ne satisfont pas les propriétés marquées «non» de cette même table.*

	una.	ano.	IUA	Juste P.	Réd. In.	conf.	ADE	IDCD
$\sum_i e_i \cdot u_i$	oui	oui	oui	**non**	**non**	oui	oui	oui
$\sum_i u_i$	oui	oui	oui	**non**	**non**	oui	**non**	oui
$\text{leximin}_i\, u_i/e_i$	oui	oui	oui	oui	oui	oui	oui	oui
$\text{leximin}_i\, u_i$	oui	oui	oui	**non**	**non**	oui	**non**	oui

Tableau 2.2 — Propriétés vérifiées par les fonctions d'utilité collective à droits exogènes inégaux.

2.4 Fonctions d'utilité collective de compromis et droits inégaux

2.4.1 Fonctions somme des puissances

Nous avons introduit au chapitre 1 deux familles de fonctions qui réalisent des compromis entre l'utilitarisme classique et l'égalitarisme pur. Nous allons nous intéresser brièvement à l'extension de la première de ces familles, somme des puissances, aux droits inégaux, en s'appuyant sur les deux fonctions de répartition que nous avons introduites (division et réplication).

Nous avons, pour le cas $u_i \div e_i = u_i/e_i$:

$$g^{(p)}_{\overrightarrow{e},div}(\overrightarrow{u}) \stackrel{def}{=} \begin{cases} sgn(p) \cdot \left(\frac{1}{m}\sum_{i=1}^{n} e_i^{1-p} \cdot u_i^p\right)^{1/p}, & p \neq 0, \\ \left(\prod_{i=1}^{n} (\frac{u_i}{e_i})^{e_i}\right)^{1/m}, & p = 0 \end{cases}$$

et, pour le cas $u_i \div e_i = u_i$:

$$g^{(p)}_{\overrightarrow{e},rep}(\overrightarrow{u}) \stackrel{def}{=} \begin{cases} sgn(p) \cdot \left(\frac{1}{m}\sum_{i=1}^{n} e_i \cdot u_i^p\right)^{1/p}, & p \neq 0, \\ \left(\prod_{i=1}^{n} u_i^{e_i}\right)^{1/m}, & p = 0 \end{cases}$$

On pourra trouver, d'un point de vue pratique, des formes plus agréables aux fonctions ci-dessus, en utilisant la propriété selon laquelle les fonctions d'utilité collective sont significatives à une transformation monotone croissante près.

Il est intéressant de caractériser ces fonctions d'utilité collective à l'aide des propriétés introduites ci-avant. Nous avons la proposition suivante :

Proposition 2.3 *La fonction d'utilité collective $g^{(p)}_{\overrightarrow{e},div}$ vérifie les propriétés suivantes pour tout p : unanimité, anonymat généralisé, IUA généralisée, conformité et IDCD. Pour tout $p < 1$, cette fonction vérifie en plus la propriété d'avantage aux droits élevés et de réduction des inégalités.*

Démonstration Les propriétés d'**unanimité** et d'**anonymat** généralisé sont immédiates. Les démonstrations pour les autres propriétés seront faites pour $p < 0$ uniquement. Les cas $p = 0$ et $p > 0$ se démontrent de manière similaire.
IUA généralisée : soit $(\overrightarrow{u}, \overrightarrow{v}, \overrightarrow{u}', \overrightarrow{v}')$ quadruplet de profils d'utilité tels que $u_i = v_i$ et $u'_i = v'_i$ pour un agent i et pour tout $k \neq i$, $u_k = u'_k$ et $v_k = v'_k$, et soit $(\overrightarrow{e}, \overrightarrow{e}')$ paire de vecteurs de droits telle que $\forall k \neq i$, $e_k = e'_k$. Alors nous avons :

$$\begin{aligned} & g^{(p)}_{\overrightarrow{e},div}(\overrightarrow{u}) \leq g^{(p)}_{\overrightarrow{e},div}(\overrightarrow{v}) \\ & \Leftrightarrow \sum_{k=1}^{n} e_k^{1-p} \cdot u_k^p \geq \sum_{k=1}^{n} e_k^{1-p} \cdot v_k^p \\ & \Leftrightarrow \cancel{e_i \frac{u_i^p}{e_i^p}} + \sum_{k=1,k\neq i}^{n} e'_k \frac{u_k'^p}{e_k'^p} \geq \cancel{e_i \frac{v_i^p}{e_i^p}} + \sum_{k=1,k\neq i}^{n} e'_k \frac{v_k'^p}{e_k'^p} \\ & \Leftrightarrow e'_i \frac{u_i'^p}{e_i'^p} + \sum_{k=1,k\neq i}^{n} e'_k \frac{u_k'^p}{e_k'^p} \geq e'_i \frac{v_i'^p}{e_i'^p} + \sum_{k=1,k\neq i}^{n} e'_k \frac{v_k'^p}{e_k'^p} \\ & \Leftrightarrow g^{(p)}_{\overrightarrow{e},div}(\overrightarrow{u}') \leq g^{(p)}_{\overrightarrow{e},div}(\overrightarrow{v}') \end{aligned}$$

Conformité : Soient $\overrightarrow{e}$ et $\overrightarrow{e}'$ deux vecteurs de droits tels que $e_k = e'_k$ pour tous $k \neq i$, et $e_i < e'_i$, et soit $(\overrightarrow{u}, \overrightarrow{v})$ une paire de profils d'utilité telle que $(g^{(p)}_{\overrightarrow{e},div}(\overrightarrow{u}) \geq g^{(p)}_{\overrightarrow{e},div}(\overrightarrow{v})$ et $u_i >$

v_i). Alors on a :

$$
\begin{aligned}
u_i > v_i \\
&\Rightarrow (e_i'^{1-p} - e_i^{1-p}) \cdot u_i^p \leq (e_i'^{1-p} - e_i^{1-p}) \cdot v_i^p \\
&\Rightarrow (e_i'^{1-p} - e_i^{1-p}) \cdot u_i^p + \left(g^{(p)}_{\overrightarrow{e},div}(\overrightarrow{u})\right)^p \\
&\qquad \leq (e_i'^{1-p} - e_i^{1-p}) \cdot v_i^p + \left(g^{(p)}_{\overrightarrow{e},div}(\overrightarrow{v})\right)^p \\
&\Rightarrow \left(g^{(p)}_{\overrightarrow{e}',div}(\overrightarrow{u})\right)^p \leq \left(g^{(p)}_{\overrightarrow{e}',div}(\overrightarrow{v})\right)^p \\
&\qquad \Rightarrow g^{(p)}_{\overrightarrow{e}',div}(\overrightarrow{u}) \geq g^{(p)}_{\overrightarrow{e}',div}(\overrightarrow{v})
\end{aligned}
$$

Avantage aux droits élevés : soient $\overrightarrow{u}$ et $\overrightarrow{v}$ deux profils d'utilité tels que $u_i = v_j$, $u_j = v_i$ et $u_k = v_k$ $\forall k \in N \setminus \{i, j\}$, avec $u_i > u_j$. Alors, on a :

$$
\begin{aligned}
e_i \geq e_j \Leftrightarrow e_i^{1-p} \geq e_j^{1-p} \Leftrightarrow e_i^{1-p} - e_j^{1-p} \geq 0 \\
&\Leftrightarrow (e_i^{1-p} - e_j^{1-p}) u_i^p \leq (e_i^{1-p} - e_j^{1-p}) u_j^p \\
&\Leftrightarrow e_i^{1-p} \cdot u_i^p + e_j^{1-p} \cdot u_j^p \leq e_i^{1-p} \cdot u_j^p + e_j^{1-p} \cdot u_i^p \\
&\Leftrightarrow e_i^{1-p} \cdot u_i^p + e_j^{1-p} \cdot u_j^p \leq e_i^{1-p} \cdot v_i^p + e_j^{1-p} \cdot v_j^p \\
&\Leftrightarrow \sum_{k=1}^{n} e_k^{1-p} \cdot u_k^p \leq \sum_{k=1}^{n} e_k^{1-p} \cdot v_k^p \\
&\qquad \Leftrightarrow g^{(p)}_{\overrightarrow{e}',div}(\overrightarrow{u}) \geq g^{(p)}_{\overrightarrow{e}',div}(\overrightarrow{v})
\end{aligned}
$$

IDCD : La propriété d'IDCD est quasiment immédiate. Multiplier tous les droits par k équivaut à multiplier l'utilité collective par k^{1-p}, qui reste un nombre positif, donc ne change pas le sens de variation de la fonction d'utilité collective.
Réduction des inégalités généralisée : Par définition, $g^{(p)}_{\overrightarrow{e}',div}$ vérifie la propriété de réduction des inégalités généralisée si et seulement si $g^{(p)}$ la vérifie au sens classique du terme, ce qui est vérifié pour tout $p < 1$. ▲

Proposition 2.4 *La fonction d'utilité collective $g^{(p)}_{\overrightarrow{e},rep}$ vérifie les propriétés suivantes pour tout p : unanimité, anonymat généralisé, IUA généralisée, conformité, avantage aux droits élevés et IDCD. En revanche, elle ne vérifie pas la propriété de réduction des inégalités.*

Démonstration La preuve de cette proposition pour le cas est assez similaire à la preuve précédente pour les propriétés d'unanimité, anonymat généralisé, IUA généralisée, conformité, avantage aux droits élevés et IDCD. Nous ne détaillerons donc pas ces cas.

Nous allons cependant montrer que quelque soit p la fonction $g^{(p)}_{\overrightarrow{e},rep}$ ne vérifie pas la propriété de réduction des inégalités. Nous nous limitons au cas $p \neq 0$ (le cas $p = 0$ est relativement similaire). Considérons un problème à deux agents de droits respectifs e_1 et e_2, tels que $e_1 > e_2$. Considérons les deux profils d'utilité $\overrightarrow{u} = (e_1 + \varepsilon, e_2 - \varepsilon)$ avec $\varepsilon > 0$, et $\overrightarrow{u}' = (e_1, e_2)$. La transformation du profil $\overrightarrow{u}$ en profil $\overrightarrow{u}'$ est clairement un transfert de Pigou-Dalton généralisé, donc si $g^{(p)}_{\overrightarrow{e},rep}$ vérifie la propriété de réduction des inégalités, on doit avoir $g^{(p)}_{\overrightarrow{e},rep}(\overrightarrow{u}') > g^{(p)}_{\overrightarrow{e},rep}(\overrightarrow{u})$.

Nous avons : $g^{(p)}_{\overrightarrow{e},rep}(\overrightarrow{u}') = \frac{1}{m}\left(e_1 \times (e_1 + \varepsilon)^p + (e_2 - \varepsilon)^p\right)^{1/p}$, et $g^{(p)}_{\overrightarrow{e},rep}(\overrightarrow{u}) = \frac{1}{m}\left(e_1^{p+1} + e_2^{p+1}\right)^{1/p}$. Or, comme $e_1 > e_2$, nous avons aussi $pe_1^p\varepsilon - pe_2^p\varepsilon > 0$, donc

$e_1^{p+1} + pe_1^p\varepsilon + e_2^{p+1} - pe_2^p\varepsilon > e_1^{p+1} + e_2^{p+1}$. En remarquant que $e_1 \times (e_1+\varepsilon)^p + (e_2-\varepsilon)^p = e_1^{p+1} + pe_1^p\varepsilon + e_2^{p+1} - pe_2^p\varepsilon + O_{\varepsilon\to 0^+}(\varepsilon)$, nous pouvons donc affirmer qu'il est possible de trouver un nombre ε suffisamment proche de 0 tel que $e_1 \times (e_1+\varepsilon)^p + (e_2-\varepsilon)^p > pe_1^{p+1} + pe_2^{p+1}$, donc au final tel que $g^{(p)}_{\overrightarrow{e},rep}(\overrightarrow{u}') < g^{(p)}_{\overrightarrow{e},rep}(\overrightarrow{u})$. Nous avons donc trouvé un transfert de Pigou-Dalton généralisé qui fait diminuer l'utilité collective, ce qui est contraire à la propriété de réduction des inégalité généralisée. ▲

2.4.2 Moyennes Pondérées Ordonnées Étendues

Si l'extension de la famille somme des puissances s'appuyant sur le principe de duplication des agents n'opposait pas de réelle difficulté, en revanche, l'extension des moyennes pondérées ordonnées est plus problématique. En effet, comment appliquer une fonction d'utilité collective de type moyenne pondérée ordonnée sur n'importe quel profil d'utilité et n'importe quel vecteur de droits exogènes, alors que la définition des moyennes pondérées ordonnées dépend du vecteur de poids, qui a une taille fixe ? La solution que nous proposons pour étendre la famille OWA afin qu'elle prenne en compte les droits inégaux est d'étendre aussi la notion de vecteur de poids :

Définition 2.23 (Vecteur de poids étendu) *Un* vecteur de poids étendu *est une fonction* $w\triangleright : \mathbb{N}_\star \to \bigcup_{k=1}^{+\infty}[0,1]^k$*, telle que :*
- $\triangleright$ $\forall n \in \mathbb{N}$, $\overrightarrow{w\triangleright(n)} \in [0,1]^n$,
- $\triangleright$ $\forall n \in \mathbb{N}$, $\sum_{i=1}^n w\triangleright(n)_i = 1$ *(condition de normalisation).*

Exemple 2.2 Nous pouvons par exemple définir une famille $w\triangleright^\alpha$ paramétrée par $\alpha > 1$ de vecteurs de poids étendus à vocation «équitable» comme suit :

$$\forall n \in \mathbb{N}_\star, \overrightarrow{w\triangleright^\alpha(n)} = \left(\frac{\alpha^{n-1}}{\sum_{i=0}^{n-1}\alpha^i}, \ldots, \frac{1}{\sum_{i=0}^{n-1}\alpha^i}\right).$$

Ainsi par exemple, si $\alpha = 2$, le vecteur de poids correspondant à 4 agents est $(8/15, 4/15, 2/15, 1/15)$.

Grâce à cette définition, nous pouvons à présent introduire la famille des moyennes pondérées ordonnées étendue :

Définition 2.24 (Famille OWA étendue) *La famille de fonctions d'utilité collective étendues* moyenne pondérée ordonnée étendue *(ou EOWA pour Extended Ordered Weighted Average) est la famille* $\{g^{w\triangleright}_{\overrightarrow{e},\div}\}$ *paramétrée par un vecteur de poids étendu* $w\triangleright$*, et tel que pour tout vecteur de droits exogènes* $\overrightarrow{e}$ *et tout profil d'utilité* $\overrightarrow{u}$ *:*

$$g^{(w\triangleright)}_{\overrightarrow{e},\div}(\overrightarrow{u}) = \sum_{i=1}^{m} w\triangleright(m)_i \times (u\triangleright\tfrac{\div}{\overrightarrow{e}})_i^{\uparrow}.$$

Exemple 2.3 (suite) Si nous reprenons le même vecteur de poids étendus que précédemment, nous aurons donc par exemple $g^{w\triangleright^2}(1,3,2,0) = 0 \times 8/15 + 1 \times 4/15 + 2 \times 2/15 + 3 \times 1/15 = 11/15$. On pourra remarquer qu'en faisant tendre α vers $+\infty$, on tend vers une fonction d'utilité collective étendue qui «représente» l'ordre leximin étendu.

2.5 Applications

Dans cette section, nous appliquons le principe de duplication des agents à quelques situations «microéconomiques» dans lesquelles apparaissent naturellement des droits exogènes. Si le choix de la fonction d'utilité collective et de la fonction de répartition sont souvent assez naturels, dans certains cas il peut être discutable. Notre point de vue n'est pas normatif (nous ne cherchons pas à imposer de solution) ; nous cherchons juste à mettre en évidence le pouvoir descriptif du schéma. Notons à nouveau que chaque fois que nous employons la fonction min, on peut bien entendu utiliser le préordre leximin en lieu et place de cette fonction.

Répartition d'un bien vital Une Organisation Non Gouvernementale doit répartir une quantité de riz entre différents pays sinistrés par la famine. Les pays (agents) sont de tailles (droits) différents. L'utilité reçue par un habitant est sa quantité de riz. La fonction de répartition est ici la division. Prenant en compte le caractère de répartition égalitariste suggéré par la nature vitale de la ressource, on conclut à la fonction d'utilité collective $\min u_i/e_i$ (allocation proportionnelle à la taille des populations).

Banqueroute (présenté en début de chapitre) Le cas relève assez clairement de la répartition d'utilité par division d'une part, et d'autre part au point de vue égalitariste sur la préférence collective. Ce qui conduit à la fonction d'utilité collective $\min_i u_i/e_i$. Si l'utilité se mesure directement en monnaie, maximiser cette fonction revient à allouer l'actif proportionnellement aux créances. C'est la solution classiquement proposée pour ce problème, mais d'autres se justifient également (voir par exemple [Young, 1994, chapitre 4]).

Constitution de comité (présenté en début de chapitre) L'utilité reçue par une circonscription est son nombre de représentants, et dans ce cas il y a une exigence égalitariste sur la préférence collective (égalité de représentation pour chaque habitant). Pour ce qui est de la fonction de répartition, la division semble la plus sensée (un représentant partage son temps entre les habitants de sa circonscription), ce qui conduit encore à la fonction d'utilité collective $\min_i u_i/e_i$, c'est-à-dire à une allocation tendant vers la proportionnalité du nombre de représentants par rapport aux populations, «tendant vers», car la difficulté de ce problème tient au fait que la proportionnalité exacte peut rarement être atteinte, du fait que le nombre de représentants est entier. Maximiser la fonction $\min_i u_i/e_i$ revient alors à une attribution des sièges selon la méthode de John Quincy Adams, dite du plus petit diviseur (voir [Balinski et Young, 2001, appendix A, proposition 3.10], qui donne aussi d'autres solutions pour ce problème).

Ressource commune avec différents investissements initiaux (présenté en début de chapitre, correspond au problème de partage de la constellation de satellites Pléiades) L'exploitation en commun d'une ressource correspond intuitivement à la division de la ressource entre les agents, et l'équité suggérée par la nature du problème implique de manière naturelle la fonction d'utilité collective égalitariste. Nous avons donc encore une fois affaire à la fonction d'utilité collective $\min_i u_i/e_i$ (allocation proportionnelle à la hauteur de l'investissement).

Productivités différentes (présenté en début de chapitre) Chaque agent est ici remplacé par un ensemble de clones tous également productifs, la production d'un agent étant la somme de la production de ses clones (fonction de répartition division). Le problème est utilitariste classique, car peu importe ce que produit chaque agent en particulier, seule la production totale compte, ce qui nous donne une fonction d'utilité collective $\sum u_i$.[6]

Prix du kWh Une compagnie distributrice d'électricité doit fixer un prix de vente du kWh d'énergie électrique pour les utilisateurs de son réseau. Ces utilisateurs sont réunis en communes (les agents), et le prix de vente fixé pour une commune constitue une désutilité (utilité négative) identique u_i pour tous les habitants de cette commune, donc la fonction de répartition entre les «clones» d'une même commune est la réplication. La répartition du coût doit être égalitariste, car il s'agit d'un bien public indispensable. La fonction d'utilité à considérer est donc $\min_i u_i$: la taille de la commune (donc le droit exogène) n'importe pas.

Infrastructures collectives Un nombre limité d'infrastructures collectives, plutôt de loisirs, doit être alloué à un certain nombre de villes (agents) ayant des populations de tailles différentes (droits). Soit k_i le nombre d'infrastructures allouées à la ville i. L'utilité de la décision k pour la ville i est $f_i(k_i)$. S'agissant d'un équipement de loisir, on peut mesurer l'utilité collective par la somme des satisfactions de chaque habitant. Si l'on admet que tous les habitants de la ville i jouissent d'une manière égale de la présence des k_i théâtres de la ville, alors l'utilité de chaque habitant (clone) est aussi $f_i(k_i)$ (réplication). Selon ce raisonnement, la fonction d'utilité collective qu'il convient de maximiser est $\sum_i (e_i \cdot u_i)$. Si maintenant l'équipement collectif n'avait pas un caractère de loisir mais de bien vital — comme un hôpital —, nous serions plutôt dans le cas (égalitarisme / réplication) et la fonction d'utilité collective convenable serait $\min_i u_i$.

La radio n groupes (nos n agents) partagent un espace commun doté d'un poste de radio pouvant diffuser n stations différentes. Les e_i (droits) membres du groupe i sont tous amateurs de la station i, et de celle-ci seulement. Il faut donc décider de la façon de partager le temps de diffusion du poste entre les n stations. Nous notons x_i la fraction de temps de diffusion dédiée à la station i ($\sum_{i=1}^{n} x_i = 1$) : nous considérerons que l'utilité de l'agent/groupe i est égale à x_i. Ici l'équité est primordiale, donc l'égalitarisme s'impose. En revanche le choix de la fonction de répartition est sujet à deux interprétations, ce qui rend l'exemple intéressant.

La première interprétation est que l'on partage du temps de «satisfaction» : un agent écoutant sa station préférée pendant un temps x_i sera satisfait à hauteur de x_i. C'est un cas de réplication, donnant la fonction d'utilité collective $\min_i u_i = \min_i x_i$: on alloue un temps de diffusion égal pour chaque station, sans se soucier du nombre d'amateurs de la station i. La seconde interprétation est que l'on partage le temps pendant lequel un groupe peut choisir sa station préférée, et dans ce cas, l'utilité x_i d'un agent est divisible entre ses clones (chaque clone peut choisir sa station préférée pendant un temps x_i/e_i) : la fonction de répartition est la division, ce qui aboutit à la fonction d'utilité collective $\min_i u_i/e_i = \min_i x_i/e_i$. Maximiser cette fonction revient à allouer un temps de diffusion proportionnel au nombre d'amateurs d'une station.

Cet exemple a été traité sans l'aide de droits exogènes inégaux dans la littérature (voir [Moulin, 2003, page 79]), de la manière suivante : les agents correspondent à l'ensemble des individus impliqués dans le partage de la radio (nos clones), l'utilité d'un agent amateur d'une station i étant la fraction x_i. Dans ce contexte, la fonction d'utilité collective utilitariste classique est difficilement justifiable : elle suggère de ne diffuser que la station qui recueille le plus d'amateurs. La fonction égalitariste est celle qui correspond à notre première solution, et résulte en un partage qui égalise le temps de diffusion de toutes les radios ($x_i = 1/n$). [Moulin, 2003] propose un compromis entre ces solutions

[6] Les droits n'ont en réalité pas «disparu», car ils apparaissent de manière cachée dans les fonctions $u_i(a_i)$. On trouve un exemple analogue dans [Moulin, 1988, page 21], traité sans l'aide des droits inégaux.

plutôt extrêmes (soit le groupe le plus nombreux impose son choix pour toute la durée de la diffusion, soit on ne tient pas du tout compte du nombre d'amateurs de chaque station), en utilisant la fonction d'utilité collective de Nash (qui s'écrit $g^\star(\overrightarrow{u}) = \prod u_i$ comme nous l'avons vu). Maximiser cette fonction revient, dans le problème de la radio, à résoudre le problème d'optimisation sous contrainte suivant : $\max_{\overrightarrow{x}} \prod_{i=1}^n x_i^{e_i}$, avec $\sum_{i=1}^n x_i = 1$. Ce problème d'optimisation classique peut être résolu analytiquement en utilisant la méthode de Lagrange, et il admet comme solution : $x_i = e_i/n$. Cette solution alloue un temps de diffusion proportionnel au nombre d'amateurs d'une station, ce qui correspond exactement à notre seconde solution (égalitarisme, division). Notons que cette solution correspond au principe de la «dictature aléatoire» : chaque agent impose son point de vue aux autres pendant une fraction $1/m$ du temps total.

2.6 Droits inégaux ordinaux

Si dans de nombreux problèmes, à l'instar de ceux présentés dans cet article, l'inégalité des droits exogènes est représentée par un vecteur de nombres de manière naturelle (différence d'investissement, populations de différentes tailles...), en revanche, dans certains autres problèmes, cette inégalité entre les agents se manifeste simplement sous la forme d'un simple ordre d'importance ou de priorité. Ainsi par exemple, dans un comité, l'avis d'un agent ayant plus d'expérience ou plus d'ancienneté comptera plus que l'avis d'un autre agent, sans qu'il ne soit vraiment possible à première vue d'associer des droits exogènes numériques cardinaux aux agents. Lors de l'allocation d'organes à des patients en attente de greffe, le choix des malades à transplanter s'effectue en fonction de listes de priorité, mais il n'est pas possible en général de transcrire ces priorités sous forme numérique [Young, 1994]. On peut envisager deux approches différentes de ce type de problèmes :

- ▷ une approche fondée sur une transcription numérique de l'ordre de priorité et une application du principe de duplication des agents au vecteur de droits obtenus par transcription ;
- ▷ une approche fondée directement sur l'ordre de priorité.

La transcription numérique d'un ordre de priorité entre les agents pose quelques problèmes, même si ce ne sont pas tout à fait les mêmes, que ceux qui concernent la transcription de préférences ordinales en préférences cardinales. La transcription numérique de l'ordre de priorité joue sur l'influence que l'on veut donner à ces droits sur le partage.

Lorsque l'on ne souhaite pas transcrire numériquement l'ordre de priorité, on peut envisager des méthodes pour l'intégrer tel quel dans le processus de décision. Un exemple de tel problème apparaît à un certain niveau dans le problème de partage de la constellation Pléiades [Lemaître *et al.*, 2004]. Dans ce problème, le partage s'effectue en deux phases : phase α, dite prioritaire, et phase β, dite routine. Ces deux phases correspondent à un ordre de priorité entre les agents, les agents militaires participant seuls à la phase α, et les agents civils participant seuls à la phase β.

Nous allons proposer dans cette section quelques méthodes intuitives pour prendre en compte des droits exogènes sous la forme d'un ordre de priorité entre les agents, dans un contexte *welfariste*. En particulier, nous allons envisager deux méthodes. Ces deux méthodes correspondent à deux visions extrêmes de la manière dont l'ordre de priorité influe sur le partage. Nous considérerons dans toute la suite qu'un ordre de priorité est un préordre total sur les agents : tous les agents sont ordonnés, mais on admet que plusieurs agents se situent au même niveau de priorité.

2.6.1 Méthode forte

La première méthode (méthode forte) considère l'ordre de priorité comme un critère prédominant dans le processus de décision collective. En particulier, ce critère prend le pas sur le critère dicté par la fonction d'utilité collective. En d'autres termes, la prise de décision se passe en plusieurs phases.

- ▷ On limite le problème aux agents de la plus haute classe de priorité, et on cherche l'ensemble de décisions qui maximisent la fonction d'utilité collective.
- ▷ Si cet ensemble ne contient qu'un élément, c'est la décision optimale (on ne tient pas du tout compte des autres agents). Sinon, on restreint l'ensemble des décisions admissibles $\mathscr{A}$ à cet ensemble de décisions optimales pour la première phase, et on maximise à nouveau la fonction d'utilité collective, en incluant cette fois-ci les agents situés au deuxième niveau de priorité.
- ▷ On raffine la sélection à chaque étape en incluant les agents de priorité directement inférieure, jusqu'à obtenir une décision unique.

Cette méthode est pertinente uniquement dans les problèmes pour lesquels les premières phases laissent de nombreuses décisions *ex-aequo*, en d'autres termes si les agents les plus prioritaires sont indifférents entre de nombreuses décisions. L'exemple typique d'un tel problème est un problème de partage dans lequel les agents ne convoitent qu'une petite partie de la ressource : après que les agents les plus prioritaires se sont partagés la partie de la ressource qu'ils convoitent, ils sont indifférents à l'attribution du reste de la ressource aux autres agents.

Exemple 2.4 Soit un problème de partage entre quatre agents. La relation de priorité entre les agents est la suivante : $\{1, 2\} \succ_{prio} \{4\} \succ_{prio} \{3\}$. On suppose que les seuls profils d'utilité correspondant à des solutions possibles sont les suivants : $(5, 5, 4, 10)$, $(2, 8, 4, 3)$, $(6, 4, 10, 5)$, $(5, 3, 8, 9)$ et $(2, 3, 18, 2)$. On suppose de plus que le critère de bien-être collectif utilisé est la somme (l'utilitarisme).

Pour calculer le meilleur profil d'utilités en utilisant l'ordre de bien-être collectif à forte priorité, on commence par restreindre la prise de décisions aux agents les plus prioritaires, c'est-à-dire 1 et 2. Les meilleurs profils pour ces deux agents sont $(5, 5, 4, 10)$, $(2, 8, 4, 3)$ et $(6, 4, 10, 5)$, produisant tous trois une utilité collective de 10 pour 1 et 2. Ensuite, on ajoute 4 (l'agent qui vient directement après dans l'ordre de priorité) pour départager ces trois profils. Le premier profil $(5, 5, 4, 10)$ est très clairement le meilleur parmi ces trois.

2.6.2 Méthode faible

La deuxième méthode possible est une méthode faible, pour laquelle l'ordre de priorité est uniquement considéré comme un critère qui départage les *ex-aequo*. Ici encore, la prise de décision se déroule en plusieurs phases :

- ▷ On cherche l'ensemble des décisions qui maximisent la fonction d'utilité collective avec tous les agents.
- ▷ S'il reste des *ex-aequo*, on enlève les agents les moins prioritaires, on limite l'ensemble des décisions admissibles à l'ensemble des décisions optimales précédentes et on cherche à maximiser la fonction d'utilité collective.
- ▷ On raffine la sélection à chaque étape en excluant les agents de priorité la plus basse, jusqu'à obtenir une décision unique.

Cette méthode est pertinente dans les problèmes pour lesquels les préférences des agents sont très différentes, et pour lesquels il existe un certain nombre de décisions optimales qu'il est impossible de départager et qui avantagent toutes des agents différents.

Exemple 2.5 (suite) On commence par comparer les profils d'utilité pour la totalité des agents. $(6, 4, 10, 5)$, $(5, 3, 8, 9)$ et $(2, 3, 18, 2)$ sont les profils qui donnent la plus forte utilité collective (25). Ensuite, on enlève 3 du partage, car c'est l'agent le moins prioritaire. Sur les deux profils restants, $(5, 3, 8, 9)$ est clairement le meilleur du point de vue de 1, 2 et 4.

On peut envisager des méthodes intermédiaires de décision entre ces deux procédés extrêmes. Nous proposons par exemple de limiter l'ensemble des décisions admissibles lors des premières phases, afin de permettre aux agents les moins prioritaires d'influer plus sur le processus de décision. Dans le cadre du partage de ressource commune, cela peut se traduire par la limitation de la quantité de ressource disponible pour le partage lors de la première phase, et l'augmentation progressive de cette limite jusqu'à partager toute la ressource lors de la dernière phase.

2.7 Conclusion sur les droits exogènes inégaux

Ce chapitre constitue le point de départ d'une réflexion générale sur la prise en compte de droits exogènes inégaux. Nous avons proposé un cadre général pour bâtir des ordres de bien-être social et des fonctions d'utilité collective prenant en compte des droits exogènes inégaux. De plus, nous avons introduit un certain nombre de fonctions d'utilité collective à droits inégaux, et caractérisé ces fonctions à l'aide de propriétés nouvellement introduites. Nous avons en outre proposé quelques pistes pour la prise en compte de droits inégaux sous forme d'ordres de priorité. Il reste de nombreux travaux à accomplir, notamment en ce qui concerne la recherche de propriétés des ordres de bien-être collectif à droits inégaux, du lien entre ces propriétés, et de la caractérisation de ces ordres de bien-être social à l'aide de ces propriétés. En outre, les pistes introduites dans le domaine des droits inégaux ordinaux restent entièrement à explorer.

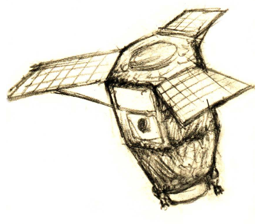
Une autre piste reste entièrement à explorer : celle de la généralisation de la fonction de répartition de manière à établir des compromis entre pure division et pure réplication. On peut se demander quel sens donner à cette généralisation de la fonction de répartition. Nous pouvons citer l'exemple suivant, en guise de point de départ. Dans l'exemple des infrastructures collectives (allocations de théâtres ou d'hôpitaux), nous avons admis que la fonction de répartition est la réplication, dans la mesure où c'est le simple fait d'exister qui permet au théâtre ou à l'hôpital d'apporter une utilité non divisée à chaque habitant. Si maintenant le théâtre ou l'hôpital est trop petit pour la ville, la fonction de répartition glisse plutôt vers la division (car tout le monde ne pourra pas en profiter en même temps quand il en a envie), et c'est à ce moment-là que des compromis pourraient être envisageables.

D'une manière plus générale, on pourrait poursuivre ces travaux sur les droits exogènes inégaux en proposant une axiomatique des ordres de bien-être social et fonctions d'utilité collective à droits exogènes inégaux, fondée par exemple sur les propriétés introduites dans ce chapitre. Une telle axiomatique pourrait éventuellement s'appuyer sur le parallèle avec les notions de dominance stochastique dans les problèmes de décision en présence de risque.

Nous n'aborderons que très peu le problème des droits exogènes inégaux dans les prochains chapitres. Cela ne sera pas nécessaire, car le modèle que nous allons définir au chapitre 3 sera assez générique pour intégrer la prise en compte de ces droits inégaux dans la fonction de répartition. Nous avons maintenant à notre disposition quelques outils formels permettant de réaliser cette intégration.

Deuxième partie

Représentation compacte et complexité



Chapitre 3

Représentation compacte

Nous avons introduit dans le chapitre 1 une modélisation formelle des principaux éléments du problème de partage. Si nous avons défini les concepts de base en termes mathématiques, en revanche, nous ne nous sommes pas préoccupé de la manière dont une instance réelle devait être représentée. Ce problème est crucial notamment en ce qui concerne la représentation des préférences des agents : une description explicite des préférences de chaque agent sur un espace d'alternatives $\mathscr{E}$ requiert soit la donnée de $|\mathscr{E}|^2$ comparaisons entre alternatives si l'on s'intéresse à des préférences ordinales, soit la donnée de $|\mathscr{E}|$ utilités si l'on s'intéresse à des préférences cardinales. Dans de nombreux problèmes réels, une telle description explicite est rédhibitoire, à cause de l'aspect combinatoire de l'espace des alternatives.

Il convient de préciser ce que l'on entend par «combinatoire». Comme nous allons le préciser dans ce chapitre, un espace combinatoire désignera un espace défini par un produit cartésien de n domaines finis de taille m : un tel espace est de taille m^n. Avoir affaire à ce genre d'espaces d'alternatives est courant en théorie de la décision : c'est concrètement le cas lorsque chaque alternative est définie par l'attribution d'une valeur particulière à un certain nombre de variables de décision, ce qui est précisément le cas en particulier dans les problèmes de partage de biens indivisibles, comme nous allons le voir.

Illustrons le phénomène d'explosion combinatoire sur deux exemples. L'exemple 3.2 nous servira de base pour illustrer certains langages de représentation compacte de préférences introduits dans ce chapitre.

Exemple 3.1 (Le repas) L'exemple du repas est un exemple classique illustrant le phénomène d'explosion combinatoire [Lang, 2004, 2006]. Considérons un agent devant exprimer ses préférences au sujet d'un repas qui se compose d'un apéritif, d'une entrée, d'un plat principal, d'un fromage, d'un dessert et d'un vin. Considérons pour simplifier qu'il y a 6 choix possibles pour chaque composante du repas (parmi lesquels un choix «vide» si l'agent désire ne rien prendre pour une composante donnée du repas). L'expression des préférences de l'agent ne pose aucun problème si celles-ci sont indépendante sur chacune des composantes du repas (par exemple si le choix sur le plat principal n'influe pas sur le choix sur la boisson) : dans ce cas, les préférences de l'agent peuvent être exprimées sous la forme de 6 structures de préférences indépendantes, ce qui ne fait au final que 36 «informations» à éliciter, si toutefois les préordres sont totaux. Cependant, face à un tel choix, un agent voudra certainement exprimer de manière naturelle des dépendances entre les variables : «si le plat principal est un poisson, je préfère un vin blanc à un vin rouge, sinon je préfère un vin rouge à un vin blanc», «s'il y a du fromage, je ne prends pas de dessert» ou encore «je n'aime pas le vin blanc sec en apéritif, mais s'il y a du foie gras en entrée je ferai une exception». Dans ce cas, l'agent devra exprimer une relation de préférence sur l'ensemble des repas différents possibles, soit

$6^6 = 46\ 656$ alternatives.

Exemple 3.2 (Héritage) Tous les problèmes de partage impliquant des biens indivisibles font apparaître de manière naturelle une structure combinatoire dès qu'il y a des dépendances préférentielles entre les objets, c'est-à-dire si certains objets sont complémentaires (avoir un ensemble de deux objets apporte une plus-value à l'agent par rapport à l'obtention des deux objets séparés) ou substituables (l'obtention de deux objets ensemble n'apporte rien à l'agent, car l'un de ces objets suffit à sa satisfaction).

Considérons par exemple un problème concernant le partage d'un héritage entre plusieurs successeurs[1]. L'héritage contient des biens immobiliers (maison, terres cultivables, parcelles boisées, ...), des véhicules (automobile, tracteur, tondeuse à gazon, ...), et un certain nombre de biens mobiliers. Si le nombre de biens à partager est m, il y a donc 2^m partages possibles (pour $m = 10$, $2^m = 1\ 024$).

Tout comme dans l'exemple du restaurant, ce nombre d'alternatives est dissuasif pour l'élicitation exhaustive des préférences, sauf si les objets sont «additivement indépendants» pour les agents, c'est-à-dire si la valeur attribuée à un couple d'objets (x, y) est égale à la somme des valeurs attribuées à chaque objet séparément. Dans ce cas, il suffit d'attribuer une utilité à chaque objet. Cependant, dans la pratique, il existe des dépendances préférentielles entre les objets : par exemple, un agent qui hérite des terres cultivables mais pas de la maison n'aura que faire de la tondeuse à gazon, mais il voudra à tout prix hériter du tracteur. Un agent qui hérite d'une cuisinière électrique ne voudra peut-être pas d'un lave-vaisselle en plus s'il habite un appartement minuscule, alors que le lave-vaisselle seul l'aurait intéressé.

Cette explosion combinatoire pose deux problèmes majeurs :

- ▷ celui de l'implantation logicielle des instances réelles, la taille de l'espace requis dépassant très rapidement la taille de l'espace mémoire disponible sur toute machine raisonnable ;
- ▷ celui de la représentation explicite de l'espace des alternatives et de l'élicitation des préférences sur cet espace, sachant que seul un être surnaturel (et doté d'une espérance de vie hors du commun) aura la faculté d'énumérer l'ensemble des alternatives ou des couples d'alternatives entre lesquels il a le choix.

Afin de pallier ces deux problèmes liés à l'explosion combinatoire, on introduit classiquement des *langages de représentation compacte*. Le rôle de ces langages est simplement de permettre une description concise d'un espace combinatoire ou d'un ensemble de structures de préférences (ordinales ou cardinales) sur un tel espace combinatoire.

Ce chapitre est organisé comme suit. Nous allons dans un premier temps nous intéresser aux moyens de représenter l'espace des alternatives en spécifiant les contraintes de manière compacte, puis nous tenterons de dresser un aperçu aussi exhaustif que possible des langages de représentation compacte qui ont été étudiés dans le domaine de l'expression des préférences, ce qui nous donnera l'occasion de discuter de la pertinence de ces langages pour le domaine des problèmes de partage. Nous nous intéresserons enfin à la représentation compacte des problèmes de partage, abordée sous deux angles différents : celui de critères purement ordinaux comme la Pareto-efficacité et l'absence d'envie appliqués à des préférences dichotomiques représentées sous forme logique, et celui de critères numériques fondés sur le *welfarisme* cardinal et une représentation logique des préférences et des contraintes.

[1] Comme nous l'avons vu au chapitre 1, il s'agit d'un problème classique dans la littérature économique sur les problèmes de partage.

3.1 Représentation compacte de l'espace des alternatives

3.1.1 Cadre formel

3.1.1.1 Espace d'alternatives combinatoires

La première question concernant la problématique de l'explosion combinatoire est liée à la représentation de l'espace des alternatives admissibles lui-même, qui est un espace combinatoire. Cependant, nous nous devons avant toute chose de préciser formellement ce que l'on entend par «espace combinatoire».

Comme nous l'avons fait remarquer en début de chapitre, l'aspect «combinatoire» d'un espace d'alternatives provient du fait qu'il est défini comme un produit cartésien de domaines finis. Nous pouvons donc définir de manière informelle un espace combinatoire comme étant un sous-ensemble en bijection avec un produit cartésien de domaines finis : dans ce contexte, chaque alternative admissible de l'espace combinatoire en question correspond à un tuple du produit cartésien des domaines. Nous prenons cependant le parti d'adopter une définition légèrement différente, et de centrer notre étude des espaces combinatoires sur la notion de *variable de décision* et d'*instanciation* :

Définition 3.1 (Variables, domaines, instanciation) *Soient $\mathscr{X} = \{\mathbf{x_i}, \ldots, \mathbf{x_n}\}$ un ensemble fini de* variable de décision*, et $\mathcal{D}$ une application de domaine qui à chaque variable $\mathbf{x_i}$ associe un ensemble fini $\mathscr{D}_{\mathbf{x_i}}$ appelé le* domaine *de $\mathbf{x_i}$. On appelle* instanciation *sur $(\mathscr{X}, \mathcal{D})$ toute application $v_{\mathscr{X},\mathcal{D}}$ qui à toute variable $\mathbf{x_i} \in \mathscr{X}$ associe une valeur $v_{\mathscr{X}}(\mathbf{x_i})$ de son domaine $\mathscr{D}_{\mathbf{x_i}} = \mathcal{D}(\mathbf{x_i})$.*

Étant donnés un ensemble de variables $\mathscr{X}$ et l'ensemble de leurs domaines défini par une application $\mathcal{D}$, on notera $Inst(\mathscr{X}, \mathcal{D})$ l'ensemble des instanciations sur $(\mathscr{X}, \mathcal{D})$.

Par la suite, nous ferons quelques simplifications de notations. Nous noterons $Inst(\mathscr{X})$ au lieu de $Inst(\mathscr{X}, \mathcal{D})$ car $\mathcal{D}$ sera toujours sous-entendu. De plus, nous noterons $v_{\mathscr{X}}$, voire v au lieu de $v_{\mathscr{X},\mathcal{D}}$, lorsque ces notations ne prêtent pas à confusion. On pourra utiliser la notation $(\mathbf{x_1} : v(\mathbf{x_1}), \ldots, \mathbf{x_n} : v(\mathbf{x_n}))$ pour désigner l'ensemble des couples (variable, valeur) correspondant à l'instanciation v, ou, lorsqu'aucune confusion n'est possible, la notation simple $(v(\mathbf{x_1}), \ldots, v(\mathbf{x_n}))$.

L'ensemble $Inst(\mathscr{X})$ des instanciations sur $\mathscr{X}$ est bien en bijection avec un produit cartésien de domaines finis, $\mathscr{D}_{\mathbf{x_1}} \times \cdots \times \mathscr{D}_{\mathbf{x_n}}$; il correspond donc à l'idée intuitive d'espace combinatoire que nous avons proposée ci-avant. C'est donc la définition que nous choisirons pour la notion d'*espace d'alternatives combinatoires* :

Définition 3.2 (Espace d'alternatives combinatoire) *Un* espace d'alternatives combinatoire *fondé sur un ensemble de variables $\mathscr{X} = \{\mathbf{x_i}, \ldots, \mathbf{x_n}\}$ et une application de domaine $\mathcal{D}$ est un ensemble $\mathscr{A}_{\mathscr{X},\mathcal{D}} \subseteq Inst(\mathscr{X}, \mathcal{D})$. Chaque élément de $\mathscr{A}_{\mathscr{X},\mathcal{D}}$ est une* alternative admissible *de l'espace.*

Tout comme pour la définition d'une instanciation, nous pourrons par la suite omettre $\mathcal{D}$ et $\mathscr{X}$ dans l'emploi de la notation $\mathscr{A}_{\mathscr{X},\mathcal{D}}$. Cette définition d'un espace d'alternatives combinatoire est intuitive et commode : toute alternative correspond à la «décision» d'affecter une valeur particulière à chaque variable. Dans l'exemple 3.1, l'ensemble des alternatives admissibles est un espace d'alternatives combinatoire fondé sur les variables $\{\mathbf{apéritif}, \mathbf{entrée}, \mathbf{plat}, \mathbf{fromage}, \mathbf{dessert}, \mathbf{vin}\}$. Les domaines de ces variables correspondent aux choix possibles, et une alternative est une application qui à chacune des variables associe une valeur de son domaine, comme par exemple l'application (**apéritif** : Macvin du Jura, **entrée** : Salade comtoise, **plat** : Saucisse de Morteau, **fromage** : Comté, **dessert** : Salade de fruits rouges, **vin** : Château l'Étoile).

Introduisons quelques notations et termes de vocabulaire supplémentaires qui nous seront utiles pour la suite. Si $\mathscr{A}$ est un espace d'alternatives combinatoires fondé sur les variables $\mathscr{X}$, alors toute instanciation v de $\mathscr{A}$ sera appelée instanciation *complète* (en plus d'être nommée alternative admissible). Une instanciation $v_{\mathscr{S}}$ ne portant que sur un sous-ensemble $\mathscr{S} \subsetneq \mathscr{X}$ sera dite *partielle*. De plus, $v_{\downarrow \mathscr{S}}$ désignera la restriction (ou projection) de l'instanciation v aux variables de $\mathscr{S}$. Étant données deux instanciations v_x et v_y sur deux sous-ensembles disjoints $\mathscr{S}_x$ et $\mathscr{S}_y$ de variables, nous noterons $\langle v_x, v_y \rangle$ l'instanciation qui attribue à toutes les variables $\mathbf{x} \in \mathscr{S}_x$ les valeurs $v_x(\mathbf{x})$, et à toutes les variables $\mathbf{y} \in \mathscr{S}_y$ les valeurs $v_y(\mathbf{y})$.

3.1.1.2 Application au partage : espace des allocations

Revenons sur le problème de partage de biens indivisibles tels qu'il a été défini au chapitre 1. Dans ce problème, l'espace des alternatives est défini comme étant l'ensemble des partages admissibles, un partage étant défini comme un ensemble de parts. Si $\mathscr{N}$ est l'ensemble des n agents et $\mathscr{O}$ l'ensemble des p objets, une alternative est donc un ensemble de n sous-ensembles de $\mathscr{O}$: c'est un élément de $\wp(\mathscr{O})^n$. L'espace des alternatives admissibles est un ensemble d'alternatives, donc un élément de $\wp(\wp(\mathscr{O})^n)$.

Il s'agit bien d'un espace combinatoire tel que nous l'avons défini. Pour s'en convaincre, nous allons exhiber un ensemble $\mathscr{X}$ de variables de décision et montrer que $\wp(\wp(\mathscr{O})^n)$ est en bijection avec l'ensemble $Inst(\mathscr{X})$. La proposition suivante est immédiate :

Proposition 3.1 (Espace des allocations) *Soient $\mathscr{N}$ un ensemble d'agents et $\mathscr{O}$ un ensemble d'objets, et soit $Alloc_{\mathscr{O},\mathscr{N}}$ l'ensemble de variables binaires $\{\mathbf{alloc(o,i)} \mid o \in \mathscr{O} \text{ et } i \in \mathscr{N}\}$. Alors $\wp(\wp(\mathscr{O})^n)$ est en bijection avec $Inst(Alloc_{\mathscr{O},\mathscr{N}})$.*

> **Démonstration** Soit h l'application qui à toute instanciation $v \in Inst(Alloc_{\mathscr{O},\mathscr{N}})$ associe le partage $h(v)$ tel que $\forall i \in \mathscr{N}$, $\pi_i = \{o \in \mathscr{O} \mid v(\mathbf{alloc(o,i)}) = 1\}$. Le partage $h(v)$ est très clairement bien défini et unique. En outre, pour tout partage $\overrightarrow{\pi}$, l'instanciation $v_{\overrightarrow{\pi}}$ telle que $\forall \mathbf{alloc(o,i)} \in Alloc_{\mathscr{O},\mathscr{N}}$, $v_{\overrightarrow{\pi}}(\mathbf{alloc(o,i)}) = 1$ si et seulement si $o \in \pi_i$ est bien définie, unique, et on peut vérifier facilement que $h(v_{\overrightarrow{\pi}}) = \overrightarrow{\pi}$. h est donc bijective. ▲

Cette proposition implique donc que tout partage peut être représenté par une instanciation v sur l'ensemble des variables $Alloc_{\mathscr{O},\mathscr{N}}$: chaque variable $\mathbf{alloc(o,i)}$ telle que $v(\mathbf{alloc(o,i)}) = 1$ correspond à l'attribution de l'objet o à l'agent i. Ainsi, tout ensemble de partages admissibles correspond à un ensemble d'instanciations. La bijection h nous fournira donc un moyen commode pour définir l'espace des alternatives admissibles.

3.1.1.3 La représentation compacte

Nous allons nous intéresser maintenant à la question de la représentation compacte. Cette question compacte concerne ici la manière de spécifier le sous-ensemble des alternatives (ou instanciations complètes) admissibles. Bien entendu, lorsque l'on parle de «description concise», que ce soit pour l'espace des alternatives ou pour la structure de préférences, il ne s'agit pas de décrire de manière compacte tous les espaces combinatoires ou toutes les structures de préférences possibles. Une telle description compacte serait impossible pour des raisons simples liées à la théorie de l'information : il y a strictement moins de 2^t formules du langage de moins de t bits. La taille caractéristique des espaces combinatoires dont nous allons parler ici est exponentielle en m^n (la taille de l'ensemble des ensembles de tuples de n éléments de domaines de taille m est 2^{m^n}). Cela prouve que certains éléments de l'espace combinatoire devront être codés par une formule de taille exponentielle en n. Ainsi, le terme de «représentation compacte» désigne l'expression concise de l'ensemble des éléments *intéressants*, ou *réalistes* de l'espace combinatoire. Naturellement, il est difficile de mettre une définition formelle derrière les termes «intéressant» ou «réaliste», dont la définition est relativement empirique : la plupart des langages d'expression compacte d'espaces combinatoires exploitent les régularités observées liées à ce que serait une description informelle et intuitive de cet espace. Ainsi par exemple, les langages à base de logique sont fondés sur le fait que la description d'un espace combinatoire par un ensemble de relations logiques entre les variables de cet espace est intuitive. Les langages à base d'enchères combinatoires fondés sur les lots (voir section 3.2.5) parient sur le fait que les agents enchérisseurs expriment naturellement leurs préférences sous la forme de mises sur des lots et de relations entre ces mises.

3.1.2 Réseaux de contraintes

Le paradigme dominant dans le domaine de la représentation d'espaces d'alternatives admissibles combinatoires est le paradigme des réseaux de contraintes [Montanari, 1974; Mackworth, 1977a]. Il est fondé sur la spécification d'un ensemble d'alternatives admissibles sous la forme de contraintes, d'une manière similaire à notre approche introduite au chapitre 1 pour définir l'espace des partages admissibles (voir la définition 1.3) :

Définition 3.3 (Réseau de contraintes) *Un réseau de contraintes est un triplet* $(\mathscr{X}, \mathcal{D}, \mathscr{C})$*, où :*
- ▷ $\mathscr{X} = \{\mathbf{x_1}, \dots, \mathbf{x_p}\}$ *est un ensemble de variables ;*
- ▷ $\mathcal{D}$ *est une application qui à toute variable* $\mathbf{x_i}$ *associe un domaine fini* $\mathscr{D}_{\mathbf{x_i}}$ *;*
- ▷ $\mathscr{C}$ *est un ensemble de contraintes, avec, pour tout* $C \in \mathscr{C}$ *:*
  - $\mathscr{X}(C) \subseteq \mathscr{X}$ *est un ensemble de variables appelé le scope de la contrainte,*
  - $\mathscr{R}(C) \subseteq Inst(\mathscr{X}(C))$ *est l'ensemble des instanciations autorisées par la contrainte.*

Une contrainte C impliquant k variables sera dite k-aire, k étant appelé l'arité de C. Si $k = 1$, la contrainte est unaire et si $k = 2$, la contrainte est dite binaire. Dans ce cadre, l'espace des alternatives admissibles est défini comme étant l'ensemble des solutions du réseau de contraintes, c'est-à-dire l'ensemble des instanciations vérifiant toutes les contraintes :

Définition 3.4 (Instanciation cohérente, solution) *Soit* C *une contrainte. Une instanciation* v *sur un ensemble de variables* $\mathscr{S} \supseteq \mathscr{X}(C)$ satisfait *la contrainte* C *si* $v_{\downarrow \mathscr{X}(C)} \in \mathscr{R}(C)$*. Sinon,* v viole C*.*

Soit $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ *un réseau de contraintes. Une instanciation* v *sur* $\mathscr{X}$ *est* cohérente *(ou* consistante*) si elle satisfait toutes les contraintes* $c \in \mathscr{C}$*. Elle est dite* incohérente *(ou* inconsistante*) sinon.*

Une solution *est une instanciation complète cohérente. Un réseau de contraintes est dit* consistant *s'il possède au moins une solution. L'ensemble des solutions d'un réseau de contraintes* $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ *sera noté* $sol(\mathscr{X}, \mathcal{D}, \mathscr{C})$*.*

Si l'on fait la supposition que pour toute contrainte C et toute instanciation v on est capable de vérifier en temps polynomial que v est autorisée par C, ce qui est une hypothèse raisonnable dans la plupart des cas, alors le problème de déterminer si un réseau de contraintes est consistant, appelé problème de satisfaction de contraintes (et sur lequel nous reviendrons au chapitre 5), est NP-complet.

Notons que la définition introduite ne précise pas explicitement comment sont spécifiés les tuples relatifs aux contraintes. S'ils sont représentés de manière explicite (c'est-à-dire par des tables associées à chaque contrainte), l'économie d'espace réalisée en utilisant ce cadre de représentation résulte du gain obtenu en décomposant la contrainte globale décrivant l'espace admissible en sous-contraintes d'arité inférieure : si le réseau comporte e contraintes d'arité maximale $r < n$, portant sur des variables dont le domaine est de taille maximale d, le coût spatial de la représentation est de ed^r, ce qui est potentiellement très inférieur à la taille de la représentation explicite d^n. Bien entendu, le gain est nul si le réseau de contraintes comporte une contrainte d'arité n.

Parmi les réseaux de contraintes, ceux qui ne comportent que des contraintes binaires ont été particulièrement étudiés dans la littérature. Il y a plusieurs raisons à cela. Tout d'abord, de nombreuses contraintes issues de problèmes réels s'expriment de manière relativement naturelle sous la forme de contraintes binaires, ce qui fait des réseaux de contraintes binaires un cadre relativement expressif. De plus, le gain obtenu en terme d'espace est relativement conséquent, ce qui est moins le cas si l'on introduit des contraintes d'arité supérieure. Cela s'explique enfin par le fait qu'il existe des algorithmes de résolution génériques très efficaces, fondés sur des mécanismes de filtrage associés aux contraintes binaires [de Givry *et al.*, 2005; Cooper et Schiex, 2004; Larrosa et Schiex, 2004; Mackworth, 1977a; Bessière et Régin, 2001]. En outre, comme le fait remarquer [Bessière, 2006], la plupart des concepts nouveaux sont beaucoup plus faciles à expliquer et à présenter si les contraintes sont binaires. On peut citer enfin un certain nombre de travaux relatifs à la traduction de contraintes non binaires en contraintes binaires (voir par exemple [Smith, 2006, page 391]).

On peut aussi, par commodité ou par économie d'espace si l'on a affaire à des contraintes non binaires, définir les contraintes *en intension* (contrairement à la forme explicite, dite *en extension*), c'est-à-dire sous la forme : $\{(\alpha_1, \ldots, \alpha_k) \mid f(\alpha_1, \ldots, \alpha_k) = 1\}$, où f est une fonction Turing-calculable à valeurs dans $\mathbb{B} = \{0, 1\}$. Dans ce cas, à l'instar des langages de représentation logique que nous allons introduire plus loin, la complexité spatiale est reportée sur la complexité temporelle liée au calcul de l'ensemble de tuples tels que $f(\alpha_1, \ldots, \alpha_k) = 1$. De nombreux travaux concernant la spécification et le développement d'algorithmes liés aux contraintes k-aires définies en intension (aussi appelées «contraintes globales») ont été effectués dans le domaine de la programmation par contraintes, qui est devenu l'un des domaines historiques de l'intelligence artificielle.

Nous reviendrons sur ces notions de réseau de contraintes et algorithmes de résolution et de filtrage dans le chapitre 5 consacré à la programmation par contraintes et au calcul de solutions leximin-optimales.

3.1.3 Variables de décision binaires

3.1.3.1 Représentation logique

Parmi les espaces d'alternatives combinatoires, ceux qui sont fondés sur des variables de décision binaires (c'est-à-dire dont le domaine de valeurs est $\mathbb{B} = \{0, 1\}$) ont une importance particulière, et ce pour plusieurs raisons. La première raison est que la binarité des variables permet l'utilisation de langages fondés sur la logique propositionnelle, qui est un langage puissant, relativement intuitif, et liée au développement historique de l'intelligence artificielle et plus précisément de la représentation des connaissances. La deuxième raison est que de nombreux problèmes réels se modélisent de manière naturelle à l'aide de variables binaires, et ce en particulier dans le domaine du partage, comme nous l'avons vu.

Introduisons avant de poursuivre quelques notions de logique propositionnelle et quelques notations qui nous seront utiles pour la suite.

Définition 3.5 (Langage propositionnel) *Soit Var un ensemble de variables propositionnelles. Le langage $\mathcal{L}_{Var}$ est le langage propositionnel fondé sur les variables propositionnelles de Var, les symboles de constantes $\bot$ et $\top$, et les connecteurs logiques $\neg$, $\vee$, et $\wedge$, c'est-à-dire tel que :*

▷ $\forall \mathbf{x} \in v$, $\mathbf{x}$ *est une formule de* $\mathcal{L}_{Var}$ *;*
▷ $\bot$ *et* $\top$ *sont des formules de* $\mathcal{L}_{Var}$ *;*
▷ $\forall \varphi_1$ *et* φ_2 *formules de* $\mathcal{L}_{Var}$*,* $\varphi_1 \vee \varphi_2$*,* $\varphi_1 \wedge \varphi_2$ *et* $\neg\varphi_1$ *sont des formules de* $\mathcal{L}_{Var}$*.*

Nous noterons de plus $\mathcal{L}^+_{Var}$ le langage $\mathcal{L}^+_{Var}$ restreint aux formules ne comportant pas de symbole $\neg$. Notons que nous n'introduisons pas les symboles d'implication $\rightarrow$ et d'équivalence $\leftrightarrow$ dans la définition du langage $\mathcal{L}_{Var}$, pour des raisons liées à la définition de $\mathcal{L}^+_{Var}$. Étant donnée une formule φ de $\mathcal{L}^+_{Var}$, nous noterons $Var(\varphi)$ l'ensemble des variables propositionnelles apparaissant dans φ.

Les définitions suivantes sont classiques en logique propositionnelle. Un *littéral* l désigne une variable propositionnelle $\mathbf{x}$ ou sa négation $\neg\mathbf{x}$. Une *clause* est une disjonction de littéraux $l_1 \vee \cdots \vee l_n$, un *cube* est une conjonction de littéraux $l_1 \wedge \cdots \wedge l_n$, une formule est en *forme normale négative* (NNF) si le symbole $\neg$ n'apparaît que devant une variable propositionnelle, elle est en *forme normale conjonctive* (CNF) si elle s'écrit comme une conjonction de clauses $Cl_1 \wedge \cdots \wedge Cl_n$, et elle est en *forme normale disjonctive* (DNF) si elle s'écrit comme une disjonction de cubes $Cu_1 \vee \cdots \vee Cu_n$.

Nous définissons de plus les notions d'interprétation et de modèle, correspondant aux notions d'instanciation et d'instanciation consistante sur des variables binaires :

Définition 3.6 (Interprétation, modèle) *Soit* $\mathcal{L}_{Var}$ *le langage propositionnel fondé sur l'ensemble de variables* Var*. Une* interprétation *sur* Var *est une instanciation* v *de* $Inst(Var)$*, c'est-à-dire une fonction de* Var *dans* $\mathbb{B}$*.*

Un modèle *de* φ *est une interprétation qui rend vraie la formule* φ *au sens de la théorie des modèles classique en logique propositionnelle. Si* v *est un modèle de* φ*, nous noterons* $v \models \varphi$*, et l'ensemble des modèles de* φ *sera noté* $Mod(\varphi)$*.*

Puisqu'à toute formule logique φ correspond un ensemble de modèles, donc un ensemble d'instanciations sur les variables propositionnelles du langage, nous pouvons donc représenter un espace d'alternatives combinatoire fondé sur des variables binaires par une formule logique :

Définition 3.7 (Représentation logique d'un espace d'alternatives) *Soit un espace combinatoire* $\mathscr{A}_{Var}$ *fondé sur des variables binaires* $Var = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$*. Une* représentation logique de l'espace $\mathscr{A}_{Var}$ *est une formule propositionnelle* $\varphi \in \mathcal{L}_{Var}$ *telle que* $Mod(\varphi) = \mathscr{A}_{Var}$*.*

Il convient de commenter quelque peu cette définition. Tout d'abord, nous pouvons remarquer que la formule représentant un ensemble de sous-ensembles d'objets n'est pas unique[2]. Pis encore, sa taille peut varier énormément : considérons par exemple la formule en forme normale conjonctive $(\mathbf{x_0^1} \vee \mathbf{x_1^1}) \wedge \cdots \wedge (\mathbf{x_0^n} \vee \mathbf{x_1^n})$. La formule en forme normale disjonctive qui lui correspond est $(\mathbf{x_0^1} \wedge \mathbf{x_0^2} \wedge \cdots \wedge \mathbf{x_0^{n-1}} \wedge \mathbf{x_0^n}) \vee (\mathbf{x_0^1} \wedge \mathbf{x_0^2} \wedge \cdots \wedge \mathbf{x_0^{n-1}} \wedge \mathbf{x_1^n}) \vee \cdots \vee (\mathbf{x_1^1} \wedge \mathbf{x_1^2} \wedge \cdots \wedge \mathbf{x_1^{n-1}} \wedge \mathbf{x_0^n}) \vee (\mathbf{x_1^1} \wedge \mathbf{x_1^2} \wedge \cdots \wedge \mathbf{x_1^{n-1}} \wedge \mathbf{x_1^n})$. Alors que la première formule a une taille de l'ordre de n, la seconde a une taille de l'ordre de $n2^n$. Il convient donc de bien choisir la formule logique pour obtenir un gain en terme d'espace.

Ensuite, il est intéressant de remarquer ici que, tout comme pour les réseaux de contraintes introduits ci-avant, le gain potentiel en espace apporté par l'expression des contraintes sous forme logique a une contrepartie en terme de temps de calcul, puisque le problème de recherche de modèle d'une formule logique quelconque (ou en forme normale conjonctive) est NP-complet d'après le théorème de Cook (voir [Cook, 1971], et le problème 19 en annexe A).

[2]Elle l'est cependant à l'équivalence logique près.

3.1.3.2 Application à l'espace des allocations

Comme nous l'avons vu, la modélisation à base de variables binaires est particulièrement bien adaptée au problème de partage, car une alternative peut être spécifiée de manière intuitive par une instanciation d'un ensemble de variables binaires $\mathbf{alloc(o, i)}$. Tout ensemble de partages peut donc être spécifié par une formule logique du langage propositionnel suivant :

Définition 3.8 (Langage $\mathcal{L}_{\mathscr{O}}^{alloc}$) *$\mathcal{L}_{\mathscr{O}}^{alloc}$ est le langage propositionnel défini par :*
- ▷ *l'ensemble $Alloc_{\mathscr{O},\mathscr{N}}$ des variables propositionnelles $\mathbf{alloc(o, i)}$ en bijection avec l'ensemble des couples objet-agent $\mathscr{O} \times \mathscr{N}$,*
- ▷ *l'ensemble des connecteurs $\{\neg, \vee, \wedge\}$.*

En d'autres termes, le langage $\mathcal{L}_{\mathscr{O}}^{alloc}$ correspond au langage $\mathcal{L}_{Alloc_{\mathscr{O},\mathscr{N}}}$. L'ensemble des partages admissibles (ou l'ensemble des contraintes d'admissibilité) pourra donc être spécifié de manière compacte grâce à la logique propositionnelle, par un ensemble de formules sur le langage

3.1.3.3 Représentation binaire de variables n-aires

Une autre raison de l'intérêt de la modélisation d'espaces combinatoires à l'aide de variables binaires est que l'on est capable de représenter n'importe quel type d'espace combinatoire à l'aide de variables binaires, moyennant l'introduction de contraintes logiques. Il suffit pour cela d'introduire un symbole propositionnel $\mathbf{egal(x_i}, \alpha)$ pour chaque couple $(\mathbf{x_i}, \alpha)$, où $\mathbf{x_i}$ est l'une des variables initiales et α une des valeurs de son domaine : une telle variable propositionnelle correspond simplement à l'affectation de la valeur α à la variable initiale $\mathbf{x_i}$. Il faut cependant assurer que les variables $\mathbf{x_i}$ ne prennent qu'une et une seule valeur parmi les α_i de leur domaine. Cela peut être spécifié à l'aide d'une contrainte logique :

$$\psi = \bigwedge_{i=1}^{n} \left(\bigvee_{\alpha \in \mathscr{D}_{\mathbf{x_i}}} \mathbf{egal}(\mathbf{x_i}, \alpha) \wedge \bigwedge_{(\alpha,\alpha') \in \mathscr{D}_{\mathbf{x_i}}^2, \alpha \neq \alpha'} \neg\mathbf{egal}(\mathbf{x_i}, \alpha) \vee \neg\mathbf{egal}(\mathbf{x_i}, \alpha') \right).$$

L'espace des alternatives admissibles peut donc être spécifié, à l'aide des symboles $(\mathbf{x_i}, \alpha)$, et d'une formule logique $\varphi = \varphi' \wedge \psi$. Cette traduction implique l'introduction de $b \times k$ variables propositionnelles, si initialement le problème était modélisé sous la forme de n variables de domaines de taille k, ainsi que l'introduction d'une formule ψ comportant de l'ordre de $n \times k^2$ variables.

On trouve quelques références concernant la traduction de variables n-aires en variables binaires [Smith, 2006, page 393], question qui a été étudiée principalement dans le contexte de la traduction du problème de satisfaction de contraintes dans le cadre du problème [SAT] [Walsh, 2000]. Bien entendu, le passage de variables n-aires aux variables binaires n'est pas sans effet en terme d'efficacité sur la résolution du problème initial.

3.1.3.4 Représentation logique et compilation de connaissances

Le succès de la logique propositionnelle dans des domaines tels que la représentation des connaissances ou l'expression compacte de domaines combinatoires a fait naturellement émerger la question de la complexité liée aux tâches de raisonnement à base de logique. Cette question est cruciale, car le fait de pouvoir exprimer des connaissances ou un espace combinatoire de manière compacte n'est pas très utile si l'on s'avère incapable de raisonner sur cette information, à cause de la trop grande complexité liée à l'expression compacte. De cette constatation est né le domaine relativement récent de la *compilation de connaissances* [Darwiche et Marquis, 2002].

La compilation de connaissances est centrée sur la notion de traduction d'une formule logique dans un langage cible, l'objectif étant de reporter une partie de la complexité de la tâche de raisonnement sur la tâche de traduction (qui peut être exécutée hors-ligne). Les trois aspects fondamentaux du langage cible d'expression de la formule logique sont les suivants : (1) sa compacité vis-à-vis du langage propositionnel initial, (2) l'ensemble des requêtes de raisonnement qui peuvent être exécutées en temps polynomial sur ce langage, et (3) l'ensemble des transformations qui peuvent être appliquées en temps polynomial sur une formule de ce langage.

Nous allons présenter succinctement quelques langages cibles dédiés à la compilation de connaissances (pour un aperçu détaillé on pourra consulter [Darwiche et Marquis, 2002], dont cette sous-section est largement inspirée). Ces langages sont tous des fragments de la logique propositionnelle, et plus précisément, la plupart des travaux sur la compilation font l'hypothèse que ces fragments sont aussi expressifs que la logique propositionnelle elle-même (c'est-à-dire que toute formule logique peut être compilée), et que ce sont des sous-ensembles du langage NNF (*Negation Normal Form*).

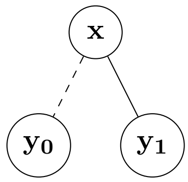
Les premiers langages considérés sont fondés sur la notion de *platitude* d'une formule du langage NNF : une formule est considérée comme plate si elle ne contient que deux niveaux d'opérateurs imbriqués. Cette propriété fournit la première grande famille de langages cibles de compilation, qui contient entre autres les f-NNF (ensemble des formules NNF plates), les CNF et les DNF.

La seconde famille de langages cibles de compilation n'impose aucune restriction sur le degré d'imbrication des opérateurs logiques, mais est fondée sur un ensemble de propriétés liées à l'agencement des littéraux et des opérateurs dans la formule. Les trois propriétés suivantes fournissent une première classe de langages :

- ▷ la *décomposabilité* : une formule φ en forme normale négative satisfait cette propriété si pour toute conjonction $\varphi_1 \wedge \varphi_2$ apparaissant dans φ, $Var(\varphi_1) \cap Var(\varphi_2) = \varnothing$ (les deux sous-formules ne partagent aucune variable) ;
- ▷ le *déterminisme* : une formule φ en forme normale négative satisfait cette propriété si pour toute disjonction $\varphi_1 \vee \varphi_2$, $\varphi_1 \wedge \varphi_2 \models \bot$ (les deux sous-formules sont contradictoires) ;
- ▷ la *régularité* (*smoothness*) : une formule φ en forme normale négative satisfait cette propriété si pour toute disjonction $\varphi_1 \vee \varphi_2$ $Var(\varphi_1) = Var(\varphi_2)$

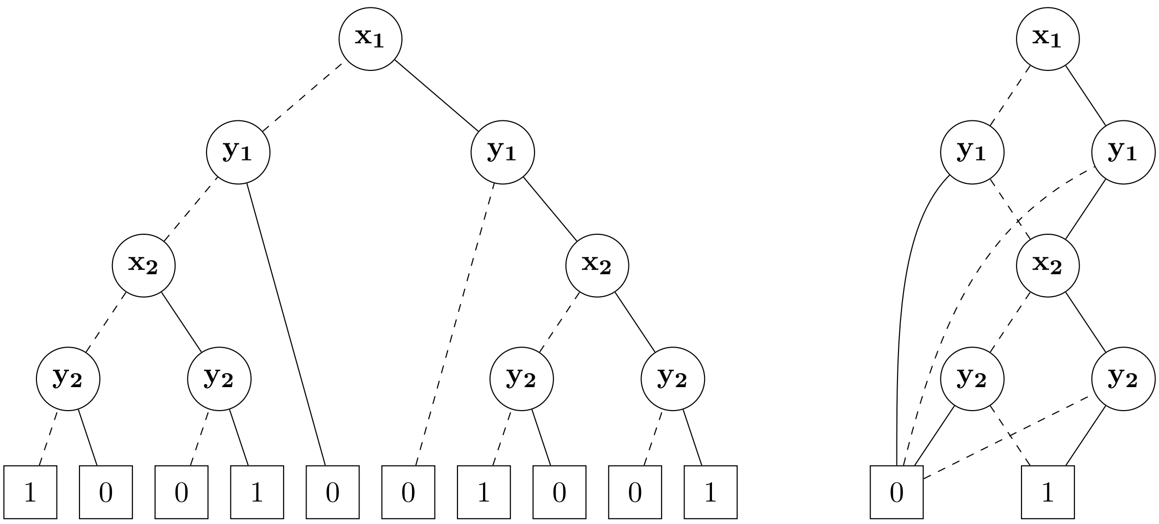
Les propriétés de décomposabilité, déterminisme et régularité définissent respectivement les classes des formules DNNF, d-NNF et s-NNF. L'ensemble des formules en forme normale négative vérifiant les propriétés de décomposabilité et déterminisme (resp. décomposabilité, déterminisme et régularité) constitue la classe d-DNNF (resp. sd-DNNF).

Nous allons terminer cette courte revue des langages de représentation de connaissances par l'un des langages les plus utilisés dans ce domaine : celui des *diagrammes de décision binaires* (ou BDD pour *Binary Decision Diagrams*). Ce langage, introduit dans [Bryant, 1986], est, tout comme les langages précédents, un sous-ensemble du langage NNF, et plus précisément il s'agit d'un sous-ensemble du langage DNNF. Il s'appuie sur une représentation logique fondée sur un opérateur particulier, l'opérateur *if-then-else* :$\mathbf{x} \mathbin{?} \mathbf{y_1} : \mathbf{y_0} = (\mathbf{x} \wedge \mathbf{y_1}) \vee (\neg\mathbf{x} \wedge \mathbf{y_0})$. Cet opérateur se représente graphiquement par un arbre enraciné en $\mathbf{x}$ et possédant deux feuilles $\mathbf{y_0}$ et $\mathbf{y_1}$. Un arc appelé *low-edge* (en pointillés) relie la racine à la feuille correspondant à la partie négative de la formule $\mathbf{y_0}$, et un arc appelé *high-edge* (en trait plein) relie la racine à la feuille correspondant à la partie positive de la formule $\mathbf{y_1}$:



Toute formule logique quelconque peut être traduite en forme normale *if-then-else* (INF), c'est-à-dire en une formule logique ne comportant que des variables propositionnelles, les constantes $\bot$ et $\top$, et des connecteurs *if-then-else* dont le premier opérande est une variable. Cette traduction s'effectue par une *expansion de Shannon*, c'est-à-dire par une succession de traductions élémentaires $\varphi \rightsquigarrow \mathbf{x} \,?\, \varphi(\mathbf{x} \leftarrow \top) : \varphi(\mathbf{x} \leftarrow \bot)$, où $\mathbf{x}$ est une variable propositionnelle de φ, et où $\varphi(\mathbf{x} \leftarrow c)$ désigne la formule φ dans laquelle on a remplacé $\mathbf{x}$ par la constante propositionnelle c. Formellement, cela revient à construire un arbre de recherche (arbre de décision) associé à la formule logique, c'est-à-dire à instancier dans un certain ordre les variables propositionnelles de φ.

Exemple 3.3 Nous allons illustrer à l'aide de l'exemple 3.2 comment une contrainte sur l'ensemble des partages possibles peut être exprimée sous la forme d'un arbre de décision, puis sous la forme d'un BDD. Considérons par exemple que pour des raisons logistiques, le tracteur (tr) et les terres cultivables (ch) forment un lot indivisible, ainsi que la tondeuse (to) et la maison (m). Une telle contrainte peut être exprimée simplement par une formule $\psi = \bigwedge_{i=1}^{n} \psi_i$, avec pour tout agent i : $\psi = (\mathbf{alloc(tr, i)} \leftrightarrow \mathbf{alloc(ch, i)}) \wedge (\mathbf{alloc(to, i)} \leftrightarrow \mathbf{alloc(m, i)})$. La formule ψ peut donc être éclatée en n contraintes différentes s'exprimant chacune par une formule du type $\varphi = (\mathbf{x_1} \leftrightarrow \mathbf{y_1}) \wedge (\mathbf{x_2} \leftrightarrow \mathbf{y_2})$. Cette formule peut être représentée par un arbre de décision comme celui de la figure 3.1(a) qui est fondé sur l'ordre $\mathbf{x_1}$, $\mathbf{y_1}$, $\mathbf{x_2}$, $\mathbf{y_2}$ sur les variables logiques.



(a) Un arbre de décision pour $(\mathbf{x_1} \leftrightarrow \mathbf{y_1}) \wedge (\mathbf{x_2} \leftrightarrow \mathbf{y_2})$

(b) Un BDD pour $(\mathbf{x_1} \leftrightarrow \mathbf{y_1}) \wedge (\mathbf{x_2} \leftrightarrow \mathbf{y_2})$ ordonné par $\mathbf{x_1} < \mathbf{y_1} < \mathbf{x_2} < \mathbf{y_2}$.

Figure 3.1 — Un arbre de décision et le diagramme de décision binaire réduit associé pour la formule $(\mathbf{x_1} \leftrightarrow \mathbf{y_1}) \wedge (\mathbf{x_2} \leftrightarrow \mathbf{y_2})$

Les diagrammes de décision binaires sont fondés sur l'arbre de décision associé à une formule et à un certain ordre des variables. La construction du BDD se fait par deux types d'opérations de réduction :

▷ en identifiant tous les sous-arbres isomorphes de l'arbre de décision initial (et en conséquence aussi tous les nœuds terminaux 0 et 1) ;
▷ en supprimant tous les nœuds ayant le même successeur pour ses deux arcs sortants *low-* et *high-edge*.

Ainsi, par exemple, dans l'arbre de la figure 3.1(a), les deux sous-arbres de racine $\mathbf{x_2}$ sont isomorphes, et peuvent donc être réduits à un seul sous arbre. On aboutit donc par ces opérations de réduction à un graphe acyclique dirigé possédant une racine et deux nœuds terminaux 0 et 1, et dont tous les nœuds possèdent deux arcs sortants. Si de plus le parcours des nœuds le long de tous les chemins se fait toujours dans le même ordre (comme c'est le cas dans le BDD de la figure 3.1(b)), le BDD est dit *ordonné* (c'est un OBDD). Si enfin le BDD ne peut plus être réduit par les opérations décrites ci-avant, le BDD est dit *réduit* (R(O)BDD).

L'intérêt principal de cette représentation, outre le gain potentiellement conséquent en terme d'espace utilisé, est fondé sur la propriété d'unicité du ROBDD : étant donné un ordre sur les variables, il n'existe qu'un seul ROBDD associé à une formule logique donnée. Cette propriété permet de tester en temps constant si une formule est insatisfiable ou s'il s'agit d'une tautologie : les ROBDD représentant ces deux formules sont des graphes ne comportant qu'un seul nœud, respectivement 0 et 1.

Pour avoir un aperçu plus détaillé des diagrammes de décision binaires, on pourra consulter l'article [Bryant, 1986] considéré comme fondateur du formalisme des BDD, ou un tutoriel proposé dans [Andersen, 1999].

3.2 Représentation compacte de préférences

3.2.1 Cadre formel

3.2.1.1 Langage de représentation compacte de préférences

Comme nous l'avons fait remarquer dans l'introduction de ce chapitre, la structure combinatoire de l'espace des alternatives rend impossible, outre la description explicite de l'ensemble des tuples formant l'espace des alternatives admissibles comme nous l'avons vu ci-avant, mais aussi et surtout la description explicite intégrale de la structure de préférences, c'est-à-dire la spécification de l'utilité de chaque alternative, ou de la relation de préférence sur tous les couples d'alternatives. La description de la structure de préférence nécessite donc l'introduction d'un *langage de représentation compacte de préférences*.

Avant d'introduire cette notion de manière formelle, nous avons besoin de quelques définitions supplémentaires. La notion de langage de représentation compacte de préférences est fondée sur un *langage formel*. Par langage formel, nous entendons de manière générale tout langage récursivement énumérable, c'est-à-dire engendré par une grammaire (pour une introduction aux langages formels, on pourra consulter l'ouvrage [Alliot *et al.*, 2002]). Cependant, la plupart des langages de représentation compacte de préférences seront fondés sur des langages classiques tels que la logique propositionnelle ou l'arithmétique de $\mathbb{N}$, ou bien seront définis par un ensemble. Un élément m d'un tel langage formel sera appelé un *mot*, ou une *formule* si l'on est dans le cas de la logique propositionnelle.

Équipés de ces notions, nous pouvons désormais introduire la définition d'un *langage de représentation compacte de préférences* :

Définition 3.9 (Langage de représentation de préférences) *Un* langage de représentation

de préférences ordinales *(resp.* cardinales*) $\mathcal{R}$ est un couple $(\mathcal{L}_\mathcal{R}, \mathcal{I}_\mathcal{R})$, où :*
- ▷ *$\mathcal{L}_\mathcal{R}$ associe à tout ensemble de variables $\mathscr{X}$ un langage formel $\mathcal{L}_\mathcal{R}(\mathscr{X})$;*
- ▷ *$\mathcal{I}_\mathcal{R}$ associe à tout ensemble de variables $\mathscr{X}$ une fonction de $\mathcal{L}_\mathcal{R}(\mathscr{X})$ dans l'ensemble des structures de préférence ordinales (resp. cardinales) sur $Inst(\mathscr{X})$.*

Pour tout mot m de $\mathcal{L}_\mathcal{R}(\mathscr{X})$ nous noterons $\succeq^m_\mathcal{R}$ (resp. $u^m_\mathcal{R}$) la structure de préférence ordinale (resp. la fonction d'utilité) $\mathcal{I}_\mathcal{R}(m)$ sur $Inst(\mathscr{X})$.

En d'autres termes, un langage de représentation de préférences est fondé sur deux composantes :
- ▷ un langage de représentation, dont la forme dépend de l'espace $Inst(\mathscr{X})$ à représenter,
- ▷ une fonction dont le rôle est de traduire n'importe quel mot (ou formule) du langage formel en la structure de préférence sur $Inst(\mathscr{X})$ à laquelle correspond ce mot.

Notons que nous considérons que l'expression des préférences se fait sur l'ensemble des alternatives $Inst(\mathscr{X})$, que celles-ci soient admissibles ou non. Cela nous permet de définir la notion de langage de représentation de préférences de manière indépendante à la définition des alternatives admissibles.

Exemple 3.4 Nous pouvons considérer par exemple un langage de représentation compacte de préférences $\mathcal{R}_{simple}$ relativement simple (nous abrégeons $\mathcal{L}_{\mathcal{R}_{simple}}$ en $\mathcal{L}_{simple}$, et $\mathcal{I}_{\mathcal{R}_{simple}}$ en $\mathcal{I}_{simple}$) défini comme suit :
- ▷ $\mathcal{L}_{simple}(\mathscr{X}) = \{(\mathscr{S}, v) \mid \mathscr{S} \subseteq \mathscr{X}$ et $v \in Inst(\mathscr{S})\}$;
- ▷ $\mathcal{I}_{simple}$ est défini comme suit : pour tout couple $(\mathscr{S}, v)$ tel que $\mathscr{S} \subseteq \mathscr{X}$ et $v \in Inst(\mathscr{S})$ et pour toute instanciation $v' \in Inst(\mathscr{X})$,
  - $u(v') = 1$ si $v'_{\downarrow \mathscr{S}} = v$;
  - $u(v') = 0$ sinon.

En d'autres termes, dans ce langage assez frustre, les préférences d'un agent sont représentées par une instanciation particulière sur un sous-ensemble $\mathscr{S}$ de variables. L'utilité de l'agent sera de 1 pour toute instanciation qui concorde avec ses préférences exprimées sur le sous-ensemble $\mathscr{S}$, et de 0 sinon. Ainsi, sur l'exemple 3.1 du repas, l'agent peut exprimer le fait qu'il veuille du vin rouge et de la salade en entrée. Tout repas qui vérifie ces préférences lui conférera une utilité égale à 1.

3.2.1.2 Application au partage : espace combinatoire d'objets

Intéressons-nous au cas particulier de l'expression des préférences dans le problème de partage de biens indivisibles. Nous avons introduit dans le chapitre 1 l'hypothèse de non exogénéité des préférences (voir section 1.2.2). Cela signifie donc concrètement que les préférences sont exprimées sur l'ensemble des parts possibles, autrement dit $\wp(\mathscr{O})$. Il s'agit encore une fois d'un espace combinatoire, qui peut être représenté à l'aide d'un ensemble de variables binaires :

Proposition 3.2 (Espace d'objets) *Soit $\mathscr{O}$ un ensemble d'objets, et soit $Var(\mathscr{O})$ l'ensemble de variables binaires $\{\mathbf{o} \mid o \in \mathscr{O}\}$. Alors $\wp(\mathscr{O})$ est en bijection avec $Inst(Var(\mathscr{O}))$ (appelé espace d'objets associé à $\mathscr{O}$).*

La preuve de ce résultat est évidente : à tout sous-ensemble $\pi \subseteq \mathscr{O}$ (nous employons la notation π afin de rester cohérent avec la définition des parts des agents utilisée dans le reste du document) est associée une interprétation de $\mathcal{L}_{Var(\mathscr{O})}$. Nous introduisons à cet effet la fonction γ, définie comme suit : $\gamma(\pi)$ est l'interprétation telle que $\gamma(\pi)(\mathbf{o}) = 1$ si et seulement si $o \in \pi$. Cette fonction est clairement bijective, donc à toute alternative v de l'espace $Inst(Var(\mathscr{O}))$ correspond un ensemble d'objets $\pi = \gamma^{-1}(v)$.

Nous pouvons remarquer que puisque l'on est capable de représenter tout sous-ensemble de $\mathscr{O}$ par une instanciation sur un ensemble de variables binaires, nous pouvons donc représenter toute part π par une formule logique :

$$\varphi(\pi) = \bigwedge_{o \in \pi} \mathbf{o} \wedge \bigwedge_{o \notin \pi} \neg \mathbf{o}.$$

Ainsi, tout ensemble de sous-ensembles de $\mathscr{O}$ peut être représenté par une formule équivalente à la disjonction des formules $\varphi(\pi)$, pour tout π faisant partie de l'ensemble. En conséquence, à tout ensemble de sous-ensembles de $\mathscr{O}$ — c'est-à-dire en termes de partage à tout ensemble de parts — correspond une formule logique φ (unique à l'équivalence logique près) dont l'ensemble des modèles correspond à l'ensemble des instanciations admissibles. Réciproquement, toute formule logique φ sur $\mathcal{L}_{Var(\mathscr{O})}$ correspond à un ensemble de parts défini par $\gamma^{-1}(Mod(\varphi))$.

Par la suite, nous emploierons quelques abus de notations pour simplifier la présentation des différentes notions introduites. La première simplification sera d'identifier les objets de $\mathscr{O}$ avec les variables propositionnelles de $Var(\mathscr{O})$ (il nous arrivera d'employer la notation mathématique simple o lorsque l'on parle d'un élément en tant qu'objet, et la notation mathématique à fonte grasse $\mathbf{o}$ lorsque l'on désignera la variable propositionnelle correspondant à o) : ainsi, nous noterons $\mathcal{L}_{\mathscr{O}}$ et $\mathcal{L}^+_{\mathscr{O}}$ les langages respectifs $\mathcal{L}_{Var(\mathscr{O})}$ et $\mathcal{L}^+_{Var(\mathscr{O})}$. De plus, nous identifierons, lorsque cela ne prête pas à confusion, l'ensemble $Inst(\mathscr{O})$ des instanciations (interprétations) sur $\mathscr{O}$ à l'ensemble $\wp(\mathscr{O})$ des sous-ensembles de $\mathscr{O}$, à l'aide de la bijection γ introduite ci-avant. Ainsi, une part $\pi \in \wp(\mathscr{O})$ sera identifiée par abus de notation à l'interprétation $\gamma(\pi) \in Inst(\mathscr{O})$. En particulier, nous emploierons la notation $\pi \models \varphi$ pour $\gamma(\pi) \models \varphi$.

Exemple 3.5 Considérons le langage logique bâti sur l'ensemble $\mathscr{O} = \{o_1, o_2, o_3, o_4, o_5\}$. L'interprétation $v = (\mathbf{o_1} : 0, \mathbf{o_2} : 1, \mathbf{o_3} : 1, \mathbf{o_4} : 0, \mathbf{o_5} : 0)$ correspond au sous-ensemble $\{o_2, o_3\}$. Il s'agit par exemple d'un modèle de la formule $\varphi = (\mathbf{o_2} \wedge \mathbf{o_3}) \vee (\mathbf{o_1} \wedge \mathbf{o_4} \wedge \mathbf{o_5})$, décrivant l'ensemble d'alternatives suivant : $\{o_2, o_3\}$, $\{o_1, o_2, o_3\}$, $\{o_1, o_4, o_5\}$, $\{o_2, o_3, o_4\}$, $\{o_2, o_3, o_5\}$, $\{o_1, o_2, o_3, o_4\}$, $\{o_1, o_2, o_3, o_5\}$, $\{o_1, o_2, o_4, o_5\}$, $\{o_2, o_3, o_4, o_5\}$, $\{o_1, o_2, o_3, o_4, o_5\}$. On a par exemple $\{o_2, o_3, o_5\} \models \varphi$.

3.2.1.3 La représentation compacte de préférences

L'étude des langages de représentation compacte de préférences est un domaine traditionnel de l'intelligence artificielle, et plus particulièrement des communautés *représentation de la connaissance et raisonnement* (KR) et *incertitude en intelligence artificielle* (UAI), simplement parce que la problématique de l'explosion combinatoire liée à l'expression de connaissances ou de préférences apparaît de manière naturelle lorsque l'on s'intéresse à la formalisation et à la simulation logicielle du raisonnement humain. Ces communautés ont introduit de nombreux langages de représentation compacte qui sont tous fondés sur l'exploitation de la structure particulière des préférences humaines. Certains de ces langages sont fondés sur la logique propositionnelle (voir la sous-section 3.2.2), partant du principe que les connaissances humaines se représentent de manière assez intuitive et efficace à l'aide de la logique. D'autres (voir entre autres les sous-section 3.2.3.2 et paragraphe 3.2.4.2) sont des modèles graphiques exploitant des propriétés locales sur l'expression des préférences.

Il est souvent difficile de juger précisément de la pertinence d'un langage de représentation compacte, et surtout de son adéquation au problème traité. Certains auteurs proposent néanmoins un certain nombre de critères plus ou moins formels (voir [Chevaleyre *et al.*, 2006a; Coste-Marquis *et al.*, 2004]) pour juger de la pertinence d'un langage $\mathcal{R}$:

- ▷ *Élicitation* : Est-il difficile d'éliciter les préférences, c'est-à-dire de construire une formule du langage $\mathcal{L}_{\mathcal{R}}$ à partir de préférences réelles ?
- ▷ *Pertinence cognitive* : Le langage $\mathcal{R}$ est-il proche de la manière dont les humains expriment leurs préférences ?

- ▷ *Puissance expressive* : Le langage $\mathcal{R}$ peut-il exprimer tous les préordres possibles, toutes les fonctions d'utilité possibles, etc.
- ▷ *Compacité relative* : Étant donnés deux langages $\mathcal{R}$ et $\mathcal{R}'$, existe-t-il pour tout élément m de $\mathcal{L}_{\mathcal{R}}$ un élément m' de $\mathcal{L}_{\mathcal{R}'}$ dont la taille est une fonction polynomiale de celle de m, et telle que $\mathcal{I}_{\mathcal{R}}(m) = \mathcal{I}_{\mathcal{R}'}(m)$?
- ▷ *Complexité computationnelle* : Est-il algorithmiquement difficile de comparer deux alternatives, de déterminer si une alternative est non-dominée, ou encore de déterminer s'il existe une alternative non dominée, vis-à-vis d'un préordre ou d'une fonction d'utilité représentés par une formule du langage $\mathcal{R}$?

Les deux premiers critères cités sont très clairement liés à la manière dont l'être humain exprime naturellement des préférences. Le problème de l'élicitation se pose en particulier dans le cas où les préférences sont numériques : s'il est facile pour un humain d'énoncer une préférence entre une bicyclette rouge et une bicyclette bleue, en revanche, il semble difficile d'affirmer que l'on préfère la bicyclette rouge à hauteur de 8.4 par rapport à la bicyclette bleue.

La puissance expressive est un critère important : il est fortement souhaitable en général que le langage puisse exprimer, si ce n'est n'importe quelle fonction d'utilité, au moins n'importe quel préordre. Pour la plupart des langages classiques présentés ci-après, ce sera le cas.

Les deux derniers critères ont respectivement trait à la complexité spatiale et temporelle des langages de représentation. Si la quantité de mémoire utilisée pour représenter un préordre est exponentielle, alors le gain espéré de la représentation compacte est perdu. De même, les langages pour lesquels la comparaison de deux alternatives est trop complexe risquent de s'avérer inutilisables en pratique pour la recherche d'une alternative optimale.

Une étude des processus d'élicitation de préférences peut être trouvée dans [Chen et Pu, 2004] ; cette question est étudiée de même dans [Sandholm et Boutilier, 2006], dans le cadre particulier des enchères combinatoires. La question de la pertinence cognitive est plus difficile à élucider et n'a été que rarement étudiée à notre connaissance dans la littérature. En revanche, les questions concernant la compacité relative et l'expressivité ont été largement étudiées dans la communauté de l'intelligence artificielle et du choix social, soit à propos de préférences ordinales [Coste-Marquis *et al.*, 2004], soit pour des préférences logiques pondérées [Chevaleyre *et al.*, 2006b], soit pour les langages de lots dans la communauté des enchères combinatoires [Boutilier et Hoos, 2001; Sandholm, 2002; Chevaleyre *et al.*, 2004; Nisan, 2006]. La complexité liée à l'expression compacte de préférences commence à être étudiée dans les mêmes communautés. On peut citer notamment [Lang, 2004] qui étudie l'impact des langages de représentation à base de logique sur la complexité du vote combinatoire, [Lipton *et al.*, 2004; Bouveret et Lang, 2005] qui s'intéressent à la complexité liée à l'absence d'envie dans le partage de ressources (nous y consacrerons une partie du chapitre 4), ou encore [Nisan, 2006; Lehmann *et al.*, 2006] qui traitent dans une moindre mesure de complexité liée à la représentation compacte de problèmes d'enchères combinatoires.

3.2.2 Modélisation à base de logique

De nombreux langages de représentation de préférences sont fondés sur la logique propositionnelle. Comme nous l'avons déjà fait remarquer, les raisons de ce succès sont diverses : nous pouvons mettre en avant le lien historiques entre la logique propositionnelle et le développement de l'intelligence artificielle, ou encore remarquer qu'il s'agit d'un langage très expressif et relativement proche de la manière dont les humains expriment naturellement leurs préférences. On pourra trouver par exemple dans [Lang, 2004, 2006] des vues d'ensemble des langages de représentation de préférences à base de logique.

Afin de ne pas perdre de vue le problème de partage et de rester cohérents avec le reste du manuscrit, nous définirons tous les langages de ce chapitre en supposant que nous avons affaire à un espace combinatoire fondé sur un ensemble d'objets $\mathscr{O}$.

3.2.2.1 Préférences dichotomiques

La plupart des langages de représentation logique de préférences sont fondés sur la notion de *but* : dans ce contexte, un but est simplement une formule logique traduisant un désir ou une préférence élémentaire. La manière la plus simple de représenter des préférences en logique propositionnelle est de ne spécifier qu'un seul but sous la forme d'une formule propositionnelle de $\mathcal{L}_{\mathscr{O}}$.

Définition 3.10 (Représentation logique d'une structure dichotomique) *Le langage de représentation logique dichotomique $\mathcal{R}_{dicho}$ est défini par :*
- ▷ $\mathcal{L}_{dicho}(\mathscr{O}) = \mathcal{L}_{\mathscr{O}}$;
- ▷ *étant donnée une formule propositionnelle φ de $\mathcal{L}_{\mathscr{O}}$, $\mathcal{I}_{dicho}(\mathscr{O})(\varphi)$ est la structure de préférence dichotomique définie par $\mathscr{G} = Mod(\varphi)$.*

En d'autres termes, une alternative (c'est-à-dire une interprétation de $\mathcal{L}_{\mathscr{O}}$) est une bonne alternative si et seulement si c'est un modèle de φ. Par exemple, si $\mathscr{O} = \{o_1, o_2, o_3\}$ et $\varphi = (\mathbf{o_1} \wedge \mathbf{o_2} \wedge \neg\mathbf{o_3}) \vee (\neg\mathbf{o_1} \wedge \mathbf{o_2} \wedge \mathbf{o_3})$, alors $\mathscr{G} = \{\{o_1, o_2\}, \{o_2, o_3\}\}$, et donc par exemple $\{o_2, o_3\}$ est préféré à $\{o_1, o_2, o_3\}$.

Il est clair que ce langage est capable d'exprimer toutes les structures de préférence dichotomiques, car tous les ensembles de sous-ensembles de $\mathscr{O}$ peuvent être représentés par une formule propositionnelle.

3.2.2.2 Préférences ordinales

Base de buts simple Le langage de représentation dichotomique étant relativement fruste, il peut être raffiné en spécifiant un ensemble de buts $GB = \{G_1, \ldots, G_n\}$ (GB pour *Goal Base*). Nous noterons $GB_{\mathscr{O}}$ une base de buts dont les formules sont dans $\mathcal{L}_{\mathscr{O}}$, et $\mathcal{GB}_{\mathscr{O}}$ le langage correspondant à l'ensemble des bases de buts sur $\mathscr{O}$. Étant donné un sous-ensemble π de $\mathscr{O}$, nous noterons de plus :
- ▷ $sat(\pi, GB_{\mathscr{O}}) = \{G_i \mid \pi \models G_i\}$;
- ▷ $nonsat(\pi, GB_{\mathscr{O}}) = \{G_i \mid \pi \not\models G_i\}$

Étant donnée une base de buts, la manière la plus basique de construire une structure de préférences ordinale est de considérer la relation d'inclusion entre les ensembles de buts satisfaits :

Définition 3.11 (Langage $\mathcal{R}_{Pareto}$) *Le langage de représentation de préférences $\mathcal{R}_{Pareto}$ est défini par :*
- ▷ $\mathcal{L}_{Pareto}(\mathscr{O}) = \mathcal{GB}_{\mathscr{O}}$;
- ▷ *étant donnée une base de buts $GB_{\mathscr{O}}$, la structure de préférences ordinale $\succeq_{Pareto}^{GB_{\mathscr{O}}}$ est telle que :*

$$\forall(\pi, \pi') \in (\wp(\mathscr{O}))^2\,,\ \pi \succeq_{Pareto}^{GB_{\mathscr{O}}} \pi' \Leftrightarrow sat(\pi, GB_{\mathscr{O}}) \supseteq sat(\pi', GB_{\mathscr{O}}).$$

Exemple 3.6 Considérons par exemple un espace d'alternatives fondé sur les objets $\mathscr{O} = \{o_1, o_2, o_3, o_4, o_5\}$, et la base de buts d'un agent $GB_{\mathscr{O}} = \{\mathbf{o_1} \vee (\mathbf{o_2} \wedge \mathbf{o_3}), \mathbf{o_4}, \mathbf{o_4} \vee \mathbf{o_5}\}$. Alors, si $\pi = \{o_1, o_4\}$ et $\pi' = \{o_2, o_3, o_5\}$ nous avons $\pi \succ_{Pareto}^{GB_{\mathscr{O}}} \pi'$, car $sat(\pi, GB_{\mathscr{O}}) = GB_{\mathscr{O}}$ et $sat(\pi', GB_{\mathscr{O}}) = \{\mathbf{o_1} \vee (\mathbf{o_2} \wedge \mathbf{o_3}), \mathbf{o_4} \vee \mathbf{o_5}\}$.

Base de buts stratifiée Le langage $\mathcal{R}_{Pareto}$ est capable de représenter tous les préordres sur l'ensemble des alternatives. Il est possible de raffiner ce langage quelque peu basique en introduisant une relation de priorité sur les formules représentant les buts :

Définition 3.12 (Base de buts stratifiée) *Une base de buts stratifiée sur $\mathscr{O}$ est un vecteur $GB_{\mathscr{O}}^{strat} = (GB_{\mathscr{O},1}, \ldots, GB_{\mathscr{O},n})$, où $GB_{\mathscr{O},i}$ est un multiensemble[3] de formules logiques de $\mathcal{L}_{\mathscr{O}}$.*

Nous noterons comme pour les bases de buts non stratifiées $\mathcal{GB}_{\mathscr{O}}^{strat}$ l'ensemble des bases de buts stratifiées sur $\mathcal{L}_{\mathscr{O}}$.

Cette notion de buts stratifiés est à la base d'un certain nombre de langages s'appuyant sur les buts et leur priorité pour définir une relation de préférence.

Définition 3.13 (Langages à base de buts stratifiés) *Les langages de représentation de préférences à base de buts stratifiés $\mathcal{R}_{best-out}$, $\mathcal{R}_{discrimin}$ et $\mathcal{R}_{leximin}$ sont tous fondés sur un langage $\mathcal{GB}_{\mathscr{O}}^{strat}$.*

Étant donnée une base de buts stratifiée $GB_{\mathscr{O}}^{strat} = (GB_{\mathscr{O},1}, \ldots, GB_{\mathscr{O},n})$ sur $\mathcal{L}_{\mathscr{O}}$, les structures de préférences induites par ces langages sont définies comme suit :

- ▷ Langage best-out *: $\pi \succeq_{bestout}^{GB_{\mathscr{O}}^{strat}} \pi'$ si et seulement si $\min\{i \mid nonsat(\pi, GB_{\mathscr{O},i}) \neq \varnothing\} \geq \min\{i \mid nonsat(\pi', GB_{\mathscr{O},i}) \neq \varnothing\}$.*
- ▷ Langage discrimin *: $\pi \succ_{discrimin}^{GB_{\mathscr{O}}^{strat}} \pi'$ si et seulement si $\exists i \leq n$ tel que $sat(\pi, GB_{\mathscr{O},i}) \supset sat(\pi', GB_{\mathscr{O},i})$ et $\forall j < i$, $sat(\pi, GB_{\mathscr{O},j}) = sat(\pi', GB_{\mathscr{O},j})$. $\pi \sim_{discrimin}^{GB_{\mathscr{O}}^{strat}} \pi'$ si et seulement si $\forall i \leq n$, $sat(\pi, GB_{\mathscr{O},i}) = sat(\pi', GB_{\mathscr{O},i})$.*
- ▷ Langage leximin *: $\pi \succ_{discrimin}^{GB_{\mathscr{O}}^{strat}} \pi'$ si et seulement si $\exists i \leq n$ tel que $|sat(\pi, GB_{\mathscr{O},i})| > |sat(\pi', GB_{\mathscr{O},i})|$ et $\forall j < i$, $|sat(\pi, GB_{\mathscr{O},j})| = |sat(\pi', GB_{\mathscr{O},j})|$. $\pi \sim_{discrimin}^{GB_{\mathscr{O}}^{strat}} \pi'$ si et seulement si $\forall i \leq n$, $|sat(\pi, GB_{\mathscr{O},i})| = |sat(\pi', GB_{\mathscr{O},i})|$.*

En d'autres termes, le langage *best-out* compare les alternatives en considérant le niveau de priorité du but non satisfait le plus prioritaire. Ce langage souffre, à l'instar de la fonction d'utilité collective min (voir chapitre 1) d'un «effet de noyade» : le langage est incapable de discriminer deux alternatives dont le plus haut niveau de priorité i du but non satisfait est identique, et inhibe l'effet de toutes les formules de niveau $j > i$ (ainsi que les autres formules de niveau i). Les critères *discrimin* et *leximin* raffinent l'ordre *best-out* sans pâtir de l'effet de noyade, en comparant par ordre de priorité les ensembles *sat* respectivement selon un critère d'inclusion et de cardinalité.

Notons que les langages $\mathcal{R}_{best-out}$ et $\mathcal{R}_{leximin}$ sont capables de générer l'ensemble des structures de préférence ordinales complètes, et que le langage $\mathcal{R}_{discrimin}$ peut engendrer l'ensemble des structures de préférence.

Exemple 3.7 Considérons toujours un espace d'alternatives fondé sur les objets $\mathscr{O} = \{o_1, o_2, o_3, o_4, o_5\}$, et une base de buts stratifiée formée des multiensembles : $GB_{\mathscr{O},1} = \{\varphi_1 = \mathbf{o_1} \vee (\mathbf{o_2} \wedge \mathbf{o_3}), \varphi_2 = \mathbf{o_4}\}$, $GB_{\mathscr{O},2} = \{\varphi_3 = \mathbf{o_5}, \varphi_4 = \neg\mathbf{o_5}, \varphi_5 = \mathbf{o_2} \wedge \mathbf{o_3}\}$ et $GB_{\mathscr{O},3} = \{\varphi_6 = \mathbf{o_4} \wedge \mathbf{o_5}\}$. Considérons les deux alternatives $\pi = \{o_1, o_4, o_5\}$ et $\pi' = \{o_2, o_3, o_4\}$. Nous avons :

	$GB_{\mathscr{O},1}$	$GB_{\mathscr{O},2}$	$GB_{\mathscr{O},3}$
$sat(\pi, .)$	φ_1, φ_2	φ_3	φ_6
$sat(\pi', .)$	φ_1, φ_2	φ_4, φ_5	$\varnothing$

[3] Un multiensemble est un ensemble dans lequel les éléments peuvent être dupliqués : $\{1, 2, 5, 2, 2\}$.

Cela nous donne les relations suivantes : $\pi \sim_{bestout}^{GB_{\mathscr{O}}^{strat}} \pi'$, π et π' sont incomparables selon $\succeq_{discrimin}^{GB_{\mathscr{O}}^{strat}}$, et $\pi' \succ_{leximin}^{GB_{\mathscr{O}}^{strat}} \pi$.

3.2.2.3 Préférences cardinales

Base de buts simple L'équivalent cardinal du langage $\mathcal{R}_{Pareto}$, fondé sur l'inclusion entre buts satisfaits, s'appuie sur le cardinal des ensembles de buts satisfaits :

Définition 3.14 (Langage $\mathcal{R}_{Card}$) *Le langage de représentation de préférences $\mathcal{R}_{Card}$ est défini par :*
- ▷ $\mathcal{L}_{Card}(\mathscr{O}) = \mathcal{GB}_{\mathscr{O}}$;
- ▷ *étant donnée une base de buts $GB_{\mathscr{O}}$, la structure de préférences ordinale $u_{Card}^{GB_{\mathscr{O}}}$ est telle que :*

$$\forall \pi \in \wp(\mathscr{O}),\ u(\pi) = |sat(\pi, GB_{\mathscr{O}})|.$$

Ce langage est capable de représenter l'ensemble des fonctions d'utilité à valeurs dans $\mathbb{N}$, d'une manière relativement basique.

L'une des extensions classiques de ce langage est fondée sur la notion de distance entre alternatives :

Définition 3.15 (pseudo-distance) *Une* pseudo-distance *est une fonction $d : \wp(\mathscr{O}) \times \wp(\mathscr{O}) \to \mathbb{N}$ telle que :*
- ▷ $\forall \pi, \pi',\ d(\pi, \pi') = 0 \Leftrightarrow \pi = \pi'$;
- ▷ $\forall \pi, \pi',\ d(\pi, \pi') = d(\pi', \pi)$.

Par exemple, la distance de Hamming est définie comme le nombre de symbole propositionnels dont la valeur diffère entre π et π'. De manière plus basique, la distance binaire est définie comme étant égale à 1 si les deux alternatives diffèrent, et 0 sinon.

Les langages de représentation logique à base de distances sont une extension directe du langage $\mathcal{R}_{Card}$. Dans ce dernier langage, on essayait de satisfaire le maximum de formules. Dans les langages à base de distances, on s'intéresse non plus à la satisfaction des formules de la base de buts, mais à la «distance» entre l'alternative considérée et ces formules (plus une alternative est «loin» des buts, moins elle est intéressante).

Définition 3.16 (Langage $\mathcal{R}_d$) *Le langage de représentation de préférences $\mathcal{R}_d$ associé à la fonction de pseudo-distance d est défini par :*
- ▷ $\mathcal{L}_d(\mathscr{O}) = \mathcal{GB}_{\mathscr{O}}$;
- ▷ *étant donnée une base de buts $GB_{\mathscr{O}} = \{G_1, \ldots, G_p\}$, la structure de préférences ordinale $u_d^{GB_{\mathscr{O}}}$ est telle que :*

$$\forall \pi \in \wp(\mathscr{O}),\ u(\pi) = f(d(\pi, G_1), \ldots, d(\pi, G_p)),$$

  où f est une fonction de $\mathbb{N}$ dans $\mathscr{V}$.

Notons que si d est la distance binaire et que f est égal à $(d_1, \ldots, d_p) \mapsto p - \sum_{i=1}^{p} d_i$, ce langage est complètement équivalent au langage $\mathcal{R}_{Card}$, puisqu'il associe à chaque alternative la même utilité que celle associée à la même alternative par $\mathcal{R}_d$.

Buts pondérés À l'instar de l'introduction de priorités pour les langages ordinaux fondés sur les bases de buts, il est possible de raffiner le langage $\mathcal{R}_{Card}$ en introduisant des poids sur les formules, assortis de fonctions d'agrégation permettant de calculer l'utilité d'une alternative à partir des formules satisfaites.

Définition 3.17 (Base de buts pondérée) *Une* base de buts pondérée *sur* $\mathcal{O}$ *est un ensemble* $GB_{\mathcal{O}}^{weighted}$ *de couples* $\delta = (\varphi(\delta), w(\delta))$*, où* $\varphi(\delta)$ *est une formule logique de* $\mathcal{L}_{\mathcal{O}}$ *et* $w(\delta)$ *une valeur dans un espace de valuation totalement ordonné* $\langle \mathscr{V}, \succeq \rangle$*.*

Nous noterons comme à l'accoutumée $\mathcal{GB}_{\mathcal{O}}^{weighted}$ l'ensemble des bases de buts pondérées sur $\mathcal{L}_{\mathcal{O}}$.

Définition 3.18 (Langage des buts pondérés) *Le langage de représentation de préférences* $\mathcal{R}_{weighted}$ *est défini par :*
- ▷ $\mathcal{L}_{weighted}(\mathcal{O}) = \mathcal{GB}_{\mathcal{O}}^{weighted}$;
- ▷ *la structure de préférences cardinale induite* $u_{weighted}^{GB_{\mathcal{O}}^{weighted}}$ *définie par :*

$$\forall \pi \in \wp(\mathcal{O}),\ u(\pi) = f_1(f_2(\{w(\delta) \mid \pi \models \varphi(\delta)\}), f_3(\{w(\delta) \mid \pi \not\models \varphi(\delta)\})),$$

où f_1*,* f_2 *et* f_3 *sont trois fonctions sur* $\mathscr{V}$*.*

Le calcul de l'utilité d'une alternative est donc calculé par agrégation séparée des poids des formules satisfaites et des poids des formules insatisfaites. Dans le langage de représentation compacte dédié au partage que nous introduirons à la toute fin de ce chapitre, la représentation des préférences des agents est fondée sur un langage à base de buts pondérés, mais dans lequel on ne prend en compte que les formules satisfaites (en d'autres termes, $f_1 = (x, y) \mapsto x$).

Ce langage de représentation est capable d'exprimer toutes les fonctions d'utilité à valeurs dans $\mathscr{V}$. La puissance expressive des restrictions (clauses, cubes, poids positifs uniquement,...) des langages à base de logique pondérée dans lesquels on ne considère que les formules satisfaites a été étudiée en détail dans [Chevaleyre *et al.*, 2006b].

Exemple 3.8 Considérons encore un espace d'alternatives fondé sur les objets $\mathcal{O} = \{o_1, o_2, o_3, o_4, o_5\}$, et la base de buts pondérée suivante : $GB_{\mathcal{O}}^{weighted} = \{\delta_1 = (\mathbf{o_1} \vee (\mathbf{o_2} \wedge \mathbf{o_3}), 5), \delta_2 = (\mathbf{o_4}, 2), \delta_3 = (\mathbf{o_5}, 2), \delta_4 = (\neg\mathbf{o_5}, 3), \delta_5 = (\mathbf{o_2} \wedge \mathbf{o_3}, 10)\}$. Nous considérons deux langages de buts pondérés : le premier est fondé sur les fonctions $+$, $+$ et $-$, et le second sur les fonctions $+$, max et $-\max$. Nous considérons les deux alternatives $\pi = \{o_1, o_4, o_5\}$ et $\pi' = \{o_2, o_3, o_4\}$, et nous notons pour chaque alternative et chaque formule pondérée le poids de la formule barré si elle n'est pas satisfaite, et non barré si elle est satisfaite :

	δ_1	δ_2	δ_3	δ_4	δ_5	$u(+,+,+)$	$u(+,\max,-\max)$
π	5	2	2	$\cancel{3}$	$\cancel{10}$	$5+2+2-3-10=-4$	$\max\{5,2,2\}-\max\{3,10\}=-5$
π'	5	2	$\cancel{2}$	3	10	$5+2-2+3+10=18$	$\max\{5,2,3,10\}-\max\{2\}=8$

Langages de compilation de connaissances Concluons cette sous-section en notant l'existence de cadres de représentation compacte de fonctions d'utilités, qui sont issus de la communauté de la compilation des connaissances :
- ▷ Les *diagrammes de décision algébriques* (ADD), permettant de représenter des fonctions pseudo-booléennes, c'est-à-dire des fonctions de $\mathbb{B}$ dans $\mathbb{N}$: le principe de représentation est assez similaire aux BDD, mais les nœuds terminaux contiennent des valeurs numériques. Pour plus de précisions on pourra se référer à [Bahar *et al.*, 1993].

▷ Les *VNNF*, qui généralisent le cadre des langages NNF au cas de fonction non exclusivement booléennes, dont la généricité permet de représenter un grand nombre de cadres de représentation compacte [Fargier et Marquis, 2007], et dont l'approche n'est pas sans rappeler celle du cadre PFU [Pralet, 2006].

3.2.3 Préférences *ceteris paribus* et CP-*nets*

3.2.3.1 Préférences *ceteris paribus*

Dans les langages de représentation de préférences à base de logique propositionnelle introduits ci-avant, les préférences sur des alternatives étaient exprimées en termes de *buts* à atteindre (ou éventuellement d'anti-buts si les pondérations étaient négatives). Si la notion de préférence entre des formules logiques pouvait y apparaître (sous la forme de priorités, ou d'utilités associées aux buts), aucune sémantique spécifique n'y était pour autant associée. L'idée des préférences *ceteris paribus* ou conditionnelles que nous allons introduire maintenant est de donner un sens précis à la notion de préférence entre formules logiques. Nous allons expliquer de façon semi-formelle de quelle manière les structures de préférence sur des alternatives peuvent être construites à partir de relations de préférence informelle sur des formules logiques.

Les logiques des préférences du type *ceteris paribus* introduites dans [Von Wright, 1963] sont fondées sur la constatation suivante : les individus expriment souvent des préférences se référant à des ensembles d'options, et non à des alternatives isolées. En terme de formules logiques, cela signifie que les préférences des agents ne sont pas représentables par des formules complètes, mais par des formules partielles. L'interprétation de la relation de préférence sur ces formules pose alors plusieurs problèmes.

▷ Que signifie exactement l'expression «la formule φ est préférée à la formule ψ» pour un agent, si φ et ψ ne sont pas mutuellement exclusives ?

▷ Comment interpréter la relation de préférence $\geq$ sur les formules en terme de relation de préférence $\succeq$ sur les alternatives ?

Répondre à la première question revient à lever des ambiguïtés liées à la non-exclusion des formules, en restreignant son application aux cas où les deux formules ne sont pas satisfaites en même temps. Considérons par exemple la préférence suivante : «je préfère le tracteur à la tondeuse à gazon» (dans le problème de l'héritage), noté tracteur$\triangleright$tondeuse. Rien ne s'oppose à ce que l'agent ne reçoive le tracteur et la tondeuse à gazon. La traduction intuitive est bien entendu la suivante : tracteur $\wedge \neg$tondeuse $>$ tondeuse $\wedge \neg$tracteur, c'est-à-dire que l'agent préfère avoir le tracteur sans la tondeuse que la tondeuse sans le tracteur. Certains cas sont légèrement plus compliqués, car une formule peut être la conséquence logique de l'autre (par exemple «je préfère avoir la tondeuse et le tracteur que la tondeuse seule» : tondeuse$\wedge$tracteur$\triangleright$tondeuse). Bien entendu, cette préférence doit être comprise comme tondeuse $\wedge$ tracteur $\triangleright$ tondeuse $\wedge \neg$tracteur. En résumé, la traduction d'une préférence $\varphi \triangleright \psi$ sur des formules non mutuellement exclusives s'interprète comme $\varphi \setminus \psi > \psi \setminus \varphi$, avec $\varphi \setminus \psi = \varphi$ si $\varphi \wedge \neg\psi$ est inconsistant, et $\varphi \wedge \neg\psi$ sinon.

Il est possible d'enrichir quelque peu la relation $>$ en tenant compte de contextes logiques. De tels contextes permettent d'exprimer des préférences du type «si j'hérite des terres cultivables, alors je préfère le tracteur à la tondeuse», qui ne sont valables que dans un contexte donné. On notera $\lambda : \varphi \triangleright \psi$, qui exprime $\lambda \wedge \varphi \triangleright \lambda \wedge \psi$. Il reste maintenant à interpréter le sens de la relation $\geq$ sur les formules logiques en terme de structure de préférence ordinale sur les alternatives. Si les alternatives concernées par les formules φ et ψ sont uniques, en d'autres termes si les formules sont complètes, l'interprétation de $\geq$ ne pose aucun problème. En revanche, si ce n'est pas le cas, il faut lever les ambiguïtés liées à la relation $\geq$. Par exemple, si l'agent dit «je préfère le tracteur à la tondeuse à gazon», il ne veut certainement pas dire « quoi qu'il arrive, je préfère le tracteur à la tondeuse à gazon ». Ainsi, si on lui laisse le choix entre hériter du tracteur seul et toucher l'intégralité de l'héritage (tondeuse y compris) sauf le tracteur, il y a fort à parier que la préférence exprimée sur le tracteur et la tondeuse ne tiendra plus.

On peut traiter ce problème en interprétant les formules « *ceteris paribus*», c'est-à-dire «toutes autres choses étant égales». Formellement, étant donnée une relation de préférence $\succeq$ sur les alternatives, et une préférence élémentaire $\varphi > \psi$ sur deux formules logiques mutuellement exclusives, $\succeq$ satisfait la relation $>$ *ceteris-paribus* si et seulement si, pour tout couple d'alternatives (π, π'), on a $\pi \succeq \pi'$ si et seulement si :

▷ $\pi \vDash \varphi$ (donc $\pi \not\vDash \psi$),

▷ et $\forall \mathbf{o} \in Var(\mathscr{O}) \setminus (Var(\varphi) \cup Var(\psi))$, $\pi \vDash \mathbf{o} \Leftrightarrow \pi' \vDash \mathbf{o}$.

En d'autres termes, une alternative est préférée à une autre relativement à une préférence $\varphi > \psi$ si et seulement si elle vérifie φ et ces deux alternatives sont identiques pour toutes les variables qui ne concernent ni φ ni ψ. Notons qu'il existe d'autres interprétations possibles pour la relation $\varphi > \psi$ que celle introduite ici. Nous ne les détaillerons pas.

On peut ainsi définir la notion de relation *ceteris paribus* induite par un ensemble de préférences GB_{CP} élémentaires du type $\varphi > \psi$ ou $\varphi \sim \psi$ comme étant l'intersection de toutes les relations $\succeq$ sur les alternatives qui satisfont les relations $>$ et $\sim$ *ceteris paribus.* Il peut ne pas exister de telle structure de préférence, s'il y a un cycle impliquant au moins une préférence stricte dans la base de buts (du type $\{\mathbf{a} > \neg\mathbf{a}, \neg\mathbf{a} > \mathbf{a}\}$). Dans ce cas la base de buts est dite *incohérente.*

Nous résumons dans le tableau ci-dessous les trois étapes de construction d'une relation de préférence *ceteris-paribus* à partir de préférences exprimées sur les formules :

Expression humaine des préférences	Clarification du flou lié à la non contradiction entre formules	Interprétation *ceteris paribus* de la relation $>$
Relation de préférence sur les formules logiques	Relation de préférence sur des formules logiques mutuellement exclusives	Relation de préférence sur les alternatives
$\varphi \rhd \psi$	$\varphi \setminus \psi > \psi \setminus \varphi$	$\pi \geq \pi'$ ssi $\pi \vDash \varphi \setminus \psi$ et $\forall o \in \mathscr{O} \setminus (Var(\psi) \cup Var(\psi))$, $\pi(o) = \pi'(o)$.

3.2.3.2 CP-*nets*

Un langage de représentation graphique de préférences ordinales fondé sur des préférences *ceteris paribus* particulières a été introduit récemment dans la communauté de l'intelligence artificielle : celui des CP-*nets* [Domshlak, 2002; Boutilier *et al.*, 2004a,b]. Ce langage est à la fois moins général que le langage des préférences *ceteris paribus*, car il se restreint à une interprétation particulière de la préférence $\rhd$, et à la fois plus général car il ne se limite pas à des préférences sur des variables binaires. La notion fondamentale sur laquelle est construit ce langage est la notion d'*indépendance préférentielle* entre variables :

Définition 3.19 (Indépendance préférentielle) *Soient* $\{\mathscr{S}_x, \mathscr{S}_y, \mathscr{S}_z\}$ *une partition de l'ensemble des variables de décision* $\mathscr{X}$ *et* $\succeq$ *une structure de préférence ordinale sur l'espace combinatoire engendré par* $\mathscr{X}$. *Alors* $\mathscr{S}_x$ *est* préférentiellement indépendant *de* $\mathscr{S}_y$ *étant donné* $\mathscr{S}_z$ *pour la relation* $\succeq$ *si et seulement si pour toute paire d'instanciations* (v_x, v'_x) *sur* $\mathscr{S}_x$, *toute paire* *d'instanciations* (v_y, v'_y) *sur* $\mathscr{S}_y$ *et toute instanciation* v_z *sur* $\mathscr{S}_z$ *:*

$$\langle v_x, v_y, v_z\rangle \succeq \langle v'_x, v_y, v_z\rangle \textit{ si et seulement si } \langle v_x, v'_y, v_z\rangle \succeq \langle v'_x, v'_y, v_z\rangle.$$

Nous rappelons que la notation $\langle v_x, v_y, v_z\rangle$ désigne l'interprétation identique à v_x sur les variables de $\mathscr{S}_x$, identique à v_y sur les variables de $\mathscr{S}_y$, et identique à v_z sur les variables de $\mathscr{S}_z$. Dans cette définition, $\mathscr{S}_z$ est interprété comme étant le contexte de la préférence (correspondant au λ introduit pour les préférences *ceteris paribus*), $\mathscr{S}_x$ est l'ensemble des variables sur lesquelles porte la préférence, et $\mathscr{S}_y$ est l'ensemble des variables dont la préférence «ne dépend pas», c'est-à-dire l'ensemble des «toutes autres choses» dans les préférences *ceteris paribus*.

Le langage des CP-*nets* est un langage de représentation graphique s'appuyant sur la notion d'indépendance préférentielle, où l'ensemble $\mathscr{S}_x$ est restreint à un singleton. Les dépendances préférentielles entre variables y sont représentées par un graphe dirigé. On pourra noter l'analogie entre ce formalisme et celui des réseaux bayésiens [Pearl, 1988].

Définition 3.20 (CP-*net*) *Un* CP-*net sur un espace combinatoire fondé sur les variables de décision* $\mathscr{X} = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$ *est un couple* $(\mathcal{G}, \mathscr{C})$*, où* $\mathcal{G}$ *est un graphe dirigé dont l'ensemble de sommets est isomorphe à* $\mathscr{X}$*, et* $\mathscr{C}$ *est un ensemble de tables de préférences conditionnelles :* $\mathscr{C} = \{C(\mathbf{x_i}) \mid \mathbf{x_i} \in \mathscr{X}\}$.

Pour tout $\mathbf{x_i} \in \mathscr{X}$ $Pa(\mathbf{x_i})$*, on note l'ensemble des variables dont les nœuds dans* $\mathcal{G}$ *sont les parents du nœud de* $\mathbf{x_i}$*. Chaque préférence conditionnelle* $C(\mathbf{x_i})$ *associe à toute instanciation* v *de* $\{\mathbf{x_j} \in Pa(\mathbf{x_i})\}$ *une relation de préférence stricte* $\succ_v^{\mathbf{x_i}}$ *sur* $\mathscr{D}_{\mathbf{x_i}}$.

Exemple 3.9 Considérons à nouveau l'exemple de l'héritage dans lequel un agent doit exprimer ses préférences sur cinq «objets» : la maison (m), les terres cultivables (ch), le tracteur (tr) et la tondeuse à gazon (to). De manière informelle, voici quelles sont les préférences de l'agent : «le tracteur ne m'intéresse pas, sauf si j'obtiens les terres cultivables», «dans ce cas-là, je ne veux pas m'encombrer à la fois d'un tracteur et d'une tondeuse à gazon : le tracteur suffira», et «la tondeuse ne m'intéresse pas en soit, sauf si j'obtiens la maison et pas le tracteur, auquel cas il faudra bien que je coupe mon herbe».

Ces préférences peuvent se traduire sous la forme d'un CP-*net* impliquant des variables binaires, présenté dans la figure 3.2. La relation de préférence induite par ce CP-*net* est présentée dans la figure 3.3.

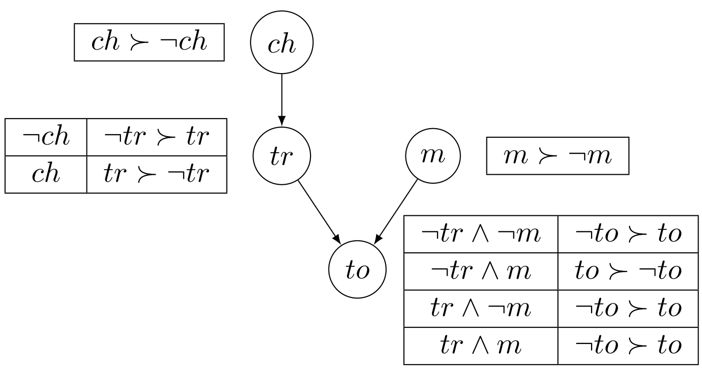


Figure 3.2 — Le CP-*net* associé aux préférences de l'exemple 3.9.

La sémantique d'un CP-*net* (c'est-à-dire la fonction $\mathcal{I}$ qui associe à un CP-*net* la structure de préférence qui lui correspond) est fondée sur la notion de relation d'ordre cohérente avec les tables :

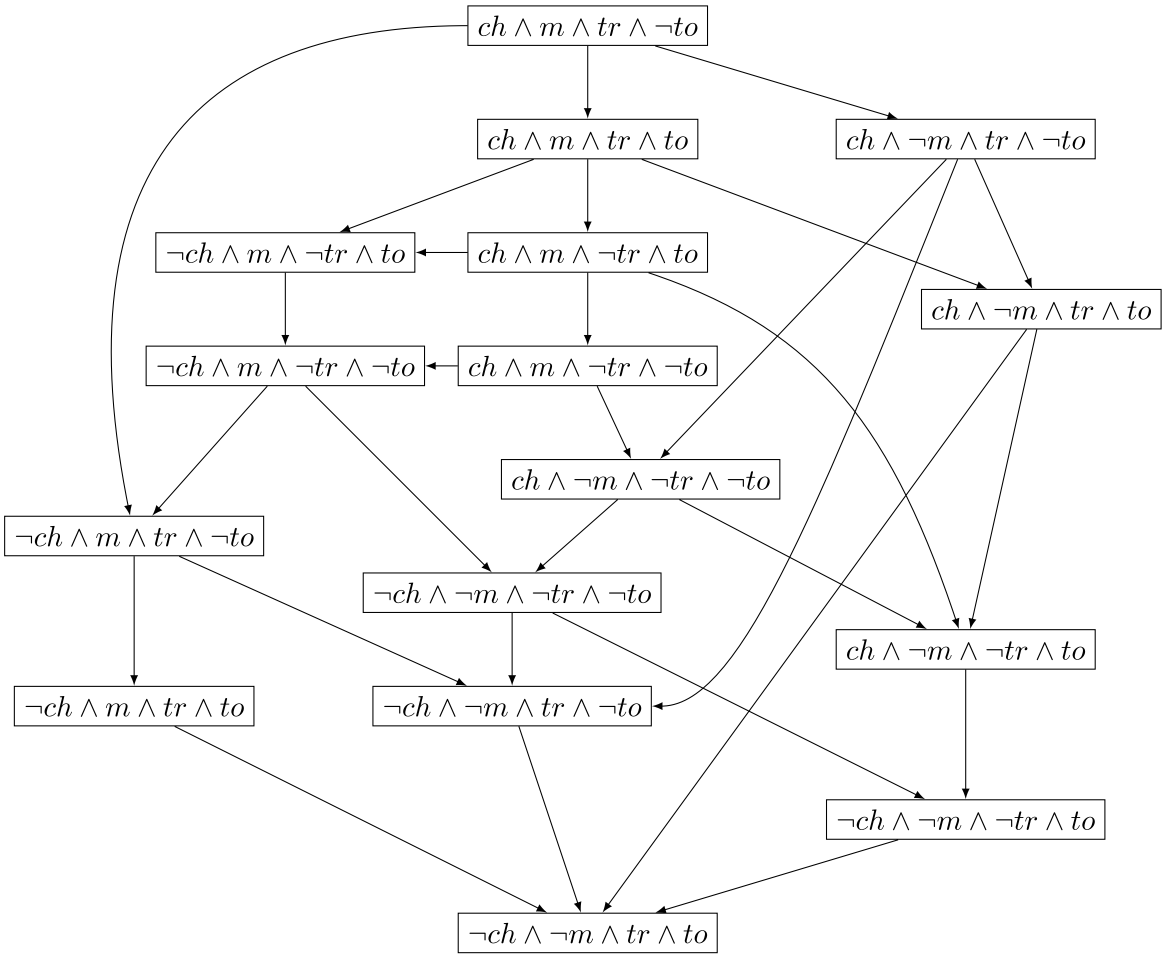


Figure 3.3 — Les préférences induites par le CP-*net* de la figure 3.2 associé aux préférences de l'exemple 3.9. La signification des arcs est la même que dans la figure 1.1 page 21 : «est préféré à».

Définition 3.21 (CP-*net* satisfiable) *Soit* $(\mathcal{G}, \mathscr{C})$ *un* CP-*net sur un espace combinatoire fondé sur les variables de décision* $\mathscr{X} = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$*, et soit* $\succ$ *une relation de préordre strict sur l'ensemble des instanciations possibles des variables de* $\mathscr{X}$*.*

La relation $\succ$ satisfait $\succ_v^{\mathbf{x_i}}$*, avec* v *une instanciation de* $Pa(\mathbf{x_i})$ *si et seulement si pour toute instanciation* v_y *de* $\mathscr{X} \setminus \left(\{\mathbf{x_i}\} \cup \bigcup_{\mathbf{x_j} \in Pa(\mathbf{x_i})} \{\mathbf{x_j}\}\right)$*, et toute paire* (v_x, v'_x) *d'instanciations de* $\mathbf{x_i}$ *on a* $\langle v_y, v_x, v \rangle \succ \langle v_y, v'_x, v \rangle$ *lorsque* $v_x(\mathbf{x_i}) \succ_v^{\mathbf{x_i}} v'_x(\mathbf{x_i})$*. La relation* $\succ$ satisfait *la table* $C(\mathbf{x_i})$ *si et seulement si elle satisfait* $\succ_v^{\mathbf{x_i}}$ *pour toute instanciation* v *de* $Pa(\mathbf{x_i})$*.*

Un CP-net est satisfiable *si et seulement s'il y a au moins une relation* $\succ$ *qui le satisfait.*

Cette définition spécifie comment les tables de préférences conditionnelles doivent être interprétées dans le graphe : pour chaque variable $\mathbf{x}$, le CP-*net* spécifie une relation de préférence portant sur son domaine, et dépendant de la valeur des parents de $\mathbf{x}$ dans le graphe. Une relation de préférence sur $\mathbf{x}$ et $Pa(\mathbf{x})$ est «compatible» avec le CP-*net* si elle respecte la relation de préférences sur $\mathscr{D}_{\mathbf{x}}$ pour toute instanciation des parents de $\mathbf{x}$.

De manière générale, il peut exister plusieurs ordres qui satisfont un CP-*net* donné. La structure de préférence induite par un CP-*net* est donc définie comme l'intersection de tous ces ordres, c'est-à-dire :

Définition 3.22 (Structure de préférence induite par un CP-*net*) *Étant donné un CP-net $(\mathcal{G}, \mathscr{C})$, la structure de préférence ordinale induite par $(\mathcal{G}, \mathscr{C})$ est telle que :*

$$v \succ_{CP-net}^{(\mathcal{G},\mathscr{C})} v' \text{ si et seulement si } v \succ v' \text{ dans toute relation } \succ \text{ satisfaisant } (\mathcal{G}, \mathscr{C}).$$

La plupart des travaux sur les CP-*nets* font l'hypothèse d'acyclicité du graphe. On peut montrer [Boutilier *et al.*, 2004a] que dans ce cas précis (acyclicité du CP-*net*), le réseau est toujours satisfiable. En outre, l'acyclicité du graphe implique des propriétés intéressantes sur la complexité des tâches associées au raisonnement sur la relation de préférence : ainsi, la comparaison d'alternatives et la recherche d'une alternative non dominée sont computationnellement raisonnables [Boutilier *et al.*, 2004b].

En revanche, le gain computationnel obtenu par ce formalisme graphique a une contrepartie en terme d'expressivité, puisque les CP-*nets* ne sont pas capables de représenter tous les préordres possibles.

Certains travaux récents s'intéressent à l'extension de ce cadre de représentation de préférences. Ainsi par exemple, [Brafman et Domshlak, 2002] introduit le cadre des TCP-*nets*, qui étendent le cadre des CP-*nets* en introduisant des importances relatives entre variables. Ce cadre est fondé sur un modèle graphique, tout comme les CP-*nets*, mais dont les arcs peuvent être de trois types différents :
- ▷ arcs de type dépendance préférentielle, dirigés et dont la sémantique est similaire à celle des arcs des CP-*nets* ;
- ▷ arcs de relation d'importance entre variables, dirigés et exprimant le fait qu'une variable est plus importante qu'une autre ;
- ▷ arcs d'importance conditionnelle, non dirigés et exprimant une relation d'importance entre variables, conditionnée par d'autres variables.

D'autres travaux se sont intéressés à la transposition des CP-*nets* dans le domaine des préférences quantitatives. Les UCP-*nets*, introduits dans [Boutilier *et al.*, 2001], sont fondés sur le même modèle graphique que les CP-*nets*, mais les tables de préférence conditionnelle contiennent des utilités au lieu de contenir des relations de préférence. Le calcul de l'utilité d'une alternative complète se fait en utilisant la notion d'indépendance additive généralisée (GAI), que nous allons introduire dans la prochaine sous-section.

3.2.3.3 Application au problème de partage

On peut légitimement s'interroger sur la pertinence des langages à base de préférences conditionnelle, et plus précisément du langage des CP-*nets* dans le cadre du partage. Nous avons montré dans l'exemple 3.9 de quelle manière les CP-*nets* peuvent être utilisés pour la représentation compacte de préférences sur des objets. Cependant, l'exemple est-il vraiment réaliste ?

La principale limitation des CP-*nets* est la restriction de la portée des tables conditionnelles à une variable unique. Cette restriction est valable dans le cas de problèmes tels que celui du repas (exemple 3.1) : un agent peut naturellement exprimer le fait qu'étant donné le choix du plat, il préfère un vin rouge à un vin blanc. Cependant, dans le cadres des problèmes de partage, l'expression d'une table de préférences conditionnelles revient à se prononcer sur l'attribution d'un objet ou non. Ce n'est pas très réaliste. Même si un agent ne désire pas un objet o, il n'exprimera probablement pas intuitivement cette préférence par $\neg o \succ o$ (en outre, dans la plupart des problèmes, l'agent peut tout simplement mettre de côté ou vendre un objet qu'il ne désire pas ; il n'y a donc pas réellement dans ce cas-là de préférence «négative»). En revanche, il sera peut-être plus enclin à exprimer le fait qu'il préfère avoir un objet o' que l'objet o.

Une telle préférence $o' \succ o$ n'est pas représentable dans le cadre strict des CP-*nets*. Cependant, cette idée d'importance relative entre la satisfaction des variables est à la base du cadre des TCP-*nets* [Brafman et Domshlak, 2002] que nous avons brièvement évoqué. Cette approche semble particulièrement intéressante pour le problème de partage. On pourrait ainsi envisager de construire un nouveau cadre de représentation compacte de préférences dédié aux problèmes d'allocations, et fondés sur les relations d'importance relative et d'importance relative conditionnelle introduites dans les TCP-*nets*. Cette ébauche de cadre semble *a priori* prometteuse.

3.2.4 Préférences additives généralisées

Si les langages de représentation de préférences introduits jusqu'ici tirent parti de la logique propositionnelle pour représenter des relations de manière compacte, les langages que nous allons introduire dans cette section utilisent des indépendances additives entre variables dans le même but[4]. L'exploitation de ces indépendances additives pour la représentation compacte de préférences a conduit aux langages k-additifs pour le partage de ressources (donc dédiés aux espaces combinatoires d'objets), et aux GAI-nets pour les espaces d'alternatives combinatoires fondés sur des variables non binaires. Nous supposerons dans toute cette sous-section (sauf spécification contraire explicite) que l'espace de valuation des utilités $\mathscr{V}$ est numérique ($\mathbb{R}$ ou $\mathbb{N}$ par exemple).

3.2.4.1 Langages de lots k-additifs

La manière la plus simple d'exprimer des utilités sur un espace d'objets sans tomber dans le piège combinatoire est de supposer une *indépendance additive* entre les objets : en d'autres termes, on suppose que $\forall(\pi_1, \pi_2) \in \wp(\mathscr{O}) \times \wp(\mathscr{O})$, $u(\pi_1 \cup \pi_2) = u(\pi_1) + u(\pi_2) - u(\pi_1 \cap \pi_2)$. De telles fonctions d'utilité sont dites *modulaires* [Chevaleyre *et al.*, 2005b; Rosenschein et Zlotkin, 1994], et peuvent être représentées uniquement par l'attribution d'une utilité à l'ensemble vide et à tous les singletons d'objets. En effet, on peut vérifier aisément le fait que, pour une fonction d'utilité modulaire u :

$$u(\pi) = u(\varnothing) + \sum_{o \in \pi} u(\{o\}).$$

Si de plus $u(\varnothing) = 0$ (ce que l'on suppose habituellement), la fonction d'utilité est dite *additive*.

Si l'hypothèse de modularité de la fonction d'utilité permet d'éviter l'explosion combinatoire, en revanche, elle exclue toute possibilité d'expression de dépendances préférentielles du type complémentarités ou substituabilités entre les variables. Afin de pallier ce défaut, la classe des langages k-additifs autorise l'expression d'utilités sur des ensembles d'objets, mais en limitant la taille maximale k des lots sur lesquels exprimer les utilités élémentaires, et en conservant l'interprétation additive de l'utilité :

Définition 3.23 (Langage k-additif) *Un langage d'expression de préférences k-additif est défini par :*

- ▷ $\mathcal{L}_{k-add}$ *: un élément de $\mathcal{L}_{k-add}(\mathscr{O})$ est un ensemble $\mathscr{M}$ de couples (π, α_π), où π est un sous-ensemble de $\mathscr{O}$ de taille au plus k, et α_π un nombre de l'espace de valuation $\mathscr{V}$;*
- ▷ $\mathcal{I}_{k-add}$ *: la fonction d'utilité $u^{\mathscr{M}}_{k-add}$ est définie par*

$$u^{\mathscr{M}}_{k-add}(\pi) = \sum_{\pi' \subseteq \pi} \alpha_{\pi'} \text{ (avec } \alpha_{\pi'} = 0 \text{ si } \pi' \text{ n'apparaît pas dans } \mathscr{M}).$$

[4] En cela, ils se rapprochent, dans un contexte légèrement différent, des formalismes fondés sur les réseaux de contrainte.

Toute fonction d'utilité exprimée sous forme explicite (*bundle form*, dans la terminologie des problèmes d'allocation de ressources), c'est-à-dire associant à chaque sous-ensemble de $\mathscr{O}$ une utilité, est exprimable dans un langage k-additif. Il suffit, pour représenter une fonction d'utilité u sous forme explicite, de calculer les coefficients α de manière inductive [Chevaleyre *et al.*, 2004] :

▷ $\alpha^u_\varnothing = u(\varnothing)$;

▷ $\alpha^u_\pi = u(\pi) - \sum_{\pi' \subsetneq \pi} \alpha^u_{\pi'}$ pour tout $\pi \subseteq \mathscr{O}$ avec $\pi \neq \varnothing$.

Les coefficients α^u_π peuvent s'exprimer sous la forme équivalente (non inductive) suivante :

$$\alpha^u_\pi = \sum_{\pi' \subsetneq \pi} (-1)^{|\pi \setminus \pi'|} u(\pi').$$

La fonction qui à toute fonction d'utilité u associe la fonction ensembliste $\pi \mapsto \alpha^u_\pi$ est appelée *transformation de Möbius*. Cette transformation, qui associe à toute fonction d'utilité sa forme k-additive, est étudiée dans le domaine des mesures floues [Grabisch, 1997] afin de réduire la complexité de représentation des mesures discrètes.

Une fonction d'utilité quelconque u est dite k-additive si sa transformée de Möbius (unique) est telle que pour tout ensemble π de taille supérieure strictement à k, on a $\alpha^u_\pi = 0$. Cette notion de transformation de Möbius met en évidence le fait qu'un langage k-additif n'est complètement expressif que si $k = |\mathscr{O}|$, car un tel langage n'est capable d'exprimer que les fonctions dont la transformée de Möbius est telle que $\alpha^u_\pi = 0$ pour tout π de taille strictement supérieure à k. À titre d'exemple, on pourra vérifier que la fonction d'utilité telle que $u(\mathscr{O}) = 1$ et $u(\pi) = 0$ pour tout $\pi \subsetneq \mathscr{O}$ n'est pas exprimable dans un langage k-additif tel que $k < |\mathscr{O}|$.

On pourra remarquer de plus qu'un langage k-additif n'est ni plus compact ni moins compact que la représentation explicite : si la représentation de certaines fonctions d'utilités requiert exponentiellement moins d'espace dans un langage k-additif que ne le requiert la représentation explicite, d'autres fonctions d'utilités exigent un espace exponentiellement plus important dans un langage k-additif que pour la représentation explicite. Les langages k-additifs ne sont donc adaptés que pour la représentation de certains types de fonctions d'utilité qui font apparaître de manière naturelle une structure additive.

Exemple 3.10 Il est possible d'exprimer des notions de complémentarité et de substituabilité entre les objets dans un langage k-additif. Revenons sur l'exemple de l'héritage, et les préférences exprimées par l'agent dans l'exemple 3.9. L'agent doit pouvoir être capable d'exprimer une complémentarité entre les terres cultivables et le tracteur, entre la maison et la tondeuse, mais en revanche une substituabilité entre le tracteur et la tondeuse (avoir les deux n'intéresse pas vraiment l'agent). L'agent commence par exprimer ses utilités sur les objets individuels, par exemple : $\alpha_{\{ch\}} = 10$, $\alpha_{\{tr\}} = 2$, $\alpha_{\{to\}} = 1$ et $\alpha_{\{m\}} = 15$. Puis il peut exprimer des complémentarités ou des substituabilités entre les objets, en donnant des utilités à des groupes d'objets : $\alpha_{\{tr,ch\}} = 8$, $\alpha_{\{tr,to\}} = -8$, $\alpha_{\{tr,m\}} = 7$, $\alpha_{\{to,m\}} = 10$.

On voit apparaître de manière naturelle les complémentarités : $\alpha_{\{to,m\}} > \alpha_{\{m\}} + \alpha_{\{to\}}$, $\alpha_{\{tr,ch\}} > \alpha_{\{tr\}} + \alpha_{\{ch\}}$. La substituabilité entre la tondeuse et le tracteur est exprimée par un poids négatif sur ce lot.

3.2.4.2 Indépendance additive généralisée, GAI-nets et CSP valués

GAI-*nets* La notion d'indépendance k-additive pour les langages impliquant des variables binaires a une correspondance pour des langages sur des domaines combinatoires impliquant des variables non binaires. La notion d'*indépendance additive généralisée* (GAI) a été appliquée de manière explicite à la représentation des utilités dans [Fishburn, 1970] et plus récemment dans [Bacchus et Grove, 1995]. L'idée est de transposer la notion d'indépendance probabiliste à la base du formalisme des réseaux bayésiens [Pearl, 1988] à la représentation des fonctions d'utilité.

Définition 3.24 (GAI-décomposition) *Soit $\mathscr{X} = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$ un ensemble de variables. Soient $\mathscr{Z}_1, \ldots, \mathscr{Z}_k$ k sous-ensembles de $[\![1, n]\!]$ tels que $\bigcup_{i=1}^k \mathscr{Z}_i = [\![1, n]\!]$.*

Une fonction d'utilité sur $Inst(\mathscr{X})$ GAI-décomposable *selon $\{\mathscr{Z}_1, \ldots, \mathscr{Z}_k\}$ est définie par la donnée de fonctions $u_i : Inst(\mathscr{Z}_i) \mapsto \mathscr{V}$, telles que :*

$$u(v) = \sum_{i=1}^{k} u_i(v_{\downarrow \mathscr{Z}_i}) \text{ pour toute instanciation } v.$$

Notons que la notion de GAI-décomposition est plus précise que la notion de k-additivité sur les lots. La k-additivité signifie simplement que l'on est capable de représenter une fonction d'utilité sur un ensemble de lots en ne spécifiant que les poids que pour des lots de taille inférieure ou égale à k. La notion de GAI-décomposition est plus forte : une fonction d'utilité GAI-décomposable est construite précisément à partir de sa décomposition en sous-fonctions.

Un modèle graphique est associé aux GAI-décompositions : le modèle des GAI-*nets* [Gonzales et Perny, 2004].

Définition 3.25 (GAI-*net*) *Soient $\mathscr{X} = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$ un ensemble de variables, et u une fonction d'utilité sur $Inst(\mathscr{X})$ GAI-décomposable selon $\{\mathscr{Z}_1, \ldots, \mathscr{Z}_k\}$. Un* GAI-*net représentant u est un graphe $\mathcal{G} = (V, E)$ tel que :*
- ▷ *$V = \{\mathscr{Z}_1, \ldots, \mathscr{Z}_k\}$;*
- ▷ *pour tout arc $(\mathscr{Z}_i, \mathscr{Z}_j) \in E$, $\mathscr{Z}_i \cap \mathscr{Z}_j \neq \varnothing$;*
- ▷ *pour tout $i \in [\![1, n]\!]$, le sous-graphe $\mathcal{G}_i = (V_i, E_i)$ induit par les sommets de $\mathcal{G}$ contenant i, défini par $V_i = \{\mathscr{Z}_j \mid i \in \mathscr{Z}_i\}$ et $E_i = \{(\mathscr{Z}_j, \mathscr{Z}_{j'}) \mid (\mathscr{Z}_j, \mathscr{Z}_{j'}) \in V_i^2 \text{ et } (\mathscr{Z}_j, \mathscr{Z}_{j'}) \in E\}$, est un sous-arbre connexe de $\mathcal{G}$ (propriété d'intersection courante).*

Les nœuds de $\mathcal{G}$ sont appelés des cliques *(ou clusters), et chaque arête $(\mathscr{Z}_i, \mathscr{Z}_j)$ est appelée* séparateur *et étiquetée avec $\mathscr{Z}_i \cap \mathscr{Z}_j$.*

Cette notion de GAI-*net* est similaire à celle de *graphe de jonction* dans les réseaux bayésiens. Dans le cas général le GAI-*net* correspondant à une fonction d'utilité GAI-décomposable n'est pas acyclique, ce qui rend les tâches liés au raisonnement sur les préférences (élicitation, agrégation de préférences...) plus difficiles. C'est pourquoi la plupart des travaux sur le sujet se restreignent à des GAI-*trees*, c'est-à-dire des GAI-*nets* acycliques.

Notons toutefois que tout GAI-*net* non acyclique peut être transformé en GAI-*tree*, comme le signale [Gonzales et Perny, 2004]. L'opération, correspondant à la transformation d'un graphe de jonction en arbre de jonction [Jensen *et al.*, 1994], procède en plusieurs étapes :
- ▷ la construction du graphe $\mathcal{G}_{dep}$ de dépendance entre variables, dont l'ensemble des sommets correspond aux variables de $\mathscr{X}$, et $(\mathbf{x_i}, \mathbf{x_j})$ est une arête de $\mathcal{G}_{dep}$ si et seulement si $\exists \mathscr{Z}_l$ tel que $\{i, j\} \subset \mathscr{Z}_l$;
- ▷ la triangulation du graphe $\mathcal{G}_{dep}$ [Robertson et Seymour, 1986], par la recherche d'un ordre d'élimination optimal (éventuellement de façon heuristique, car il s'agit d'un problème NP-difficile) ;
- ▷ la construction des nouvelles cliques à partir du graphe triangulé.

On pourra aussi noter l'existence d'un algorithme de construction d'un arbre de jonction qui ne fait pas appel à la triangulation du graphe de dépendances, mais à des recherches successives de coupes minimales [Koster, 1999]. Enfin, nous pouvons citer les travaux [Jégou et Terrioux, 2003;

de Givry *et al.*, 2006] issus de la communauté des problèmes de satisfaction de contraintes, qui sont très proches des travaux sur les GAI-*nets* en ceci qu'ils exploitent une décomposition arborescente du graphe de contraintes associé à un réseau de contraintes pour faciliter la résolution du problème. La notion de décomposition arborescente est formellement identique à la notion de transformation d'un graphe de jonction (ou d'un GAI-*net*) en arbre de jonction (ou GAI-*tree*).

Exemple 3.11 Si le modèle des GAI-*nets* est un formalisme dédié aux variables non exclusivement binaires, il peut très bien s'appliquer en particulier aux variables binaires, et donc aux problèmes de partage. Revenons sur le problème de l'héritage, l'agent ayant les mêmes préférences que dans l'exemple 3.9. Ces préférences sont naturellement GAI-décomposables, exactement de la même manière qu'elles sont exprimables dans un langage 2-additif. Les utilités associées aux objets et aux 2-lots de l'exemple 3.10 fournissent directement les tables correspondant aux relations GAI (unaires et binaires) :

ch	0	1
u_1	0	10

m	0	1
u_2	0	15

tr	0	1
u_3	0	2

to	0	1
u_4	0	1

ch	0	0	1	1
tr	0	1	0	1
u_5	0	0	0	8

tr	0	0	1	1
to	0	1	0	1
u_6	0	0	0	-8

m	0	0	1	1
tr	0	1	0	1
u_7	0	0	0	7

m	0	0	1	1
to	0	1	0	1
u_8	0	0	0	10

Ces préférences produisent par exemple le GAI-*net* présenté sur la figure 3.4 (les nœuds circulaires correspondent aux clusters, et les nœuds rectangulaires sont les séparateurs entre les clusters). Ce GAI-*net* peut être transformé en le GAI-*tree* présenté sur la même figure, par triangulation et calcul des cliques maximales.

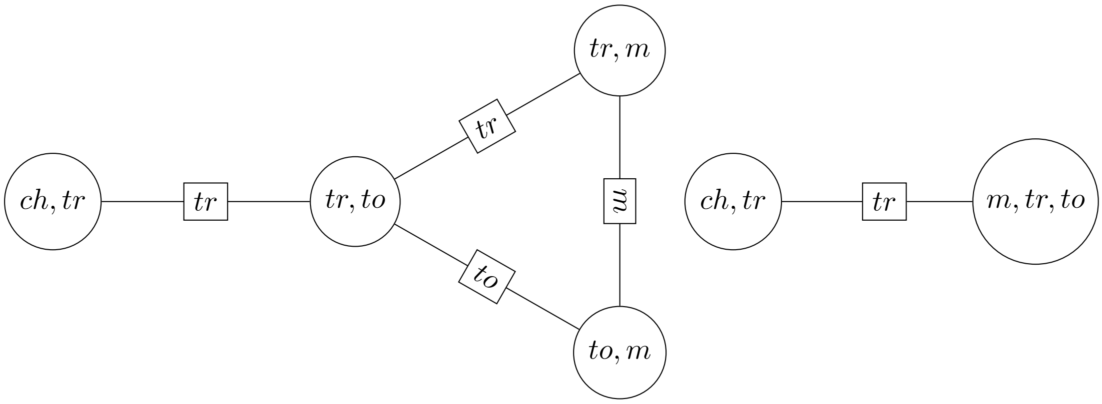


Figure 3.4 — Le GAI-*net* et son GAI-*tree* correspondant pour l'exemple 3.11.

Pour clore cette brève introduction aux GAI-*nets*, nous pouvons noter l'existence de travaux récents portant sur la transcription de structures de préférences exprimées dans un langage compact ordinal en fonctions d'utilités additives généralisées. Nous pouvons citer par exemple [Brafman *et al.*, 2004], qui introduit un certain nombre de résultats sur la transcriptions de TCP-*nets* sous la forme numérique additive généralisée.

CSP valués La notion d'indépendance additive généralisée et l'outil graphique des GAI-*nets* ont de nombreuses similitudes avec un cadre de représentation de problèmes combinatoires que nous avons déjà évoqué : celui des problèmes de satisfaction de contraintes. En effet, comme nous l'avons signalé précédemment, le fait d'exprimer des contraintes portant sur des sous-ensembles de variables correspond à une décomposition de l'espace combinatoire en plusieurs entités indépendantes (dans le sens où la satisfaction d'une contrainte particulière sur un sous-ensemble n'a aucune influence sur la satisfaction d'une autre contrainte si cette autre contrainte ne partage aucune variable avec la première).

Jusqu'à peu, l'idée de préférence était complètement absente du cadre des problèmes de satisfaction de contraintes, mais l'extension de ce cadre aux contraintes valuées a comblé ce manque, en permettant l'expression de préférences sur la satisfaction des différentes contraintes. Cette extension a mené à l'introduction de deux cadres différents [Bistarelli *et al.*, 1999] : les *Semiring*-CSP [Bistarelli *et al.*, 1995, 1997], et les CSP valués (VCSP) [Schiex *et al.*, 1995, 1997]. Nous présentons très rapidement le deuxième cadre.

Définition 3.26 (Réseau de contraintes valué) *Un* réseau de contraintes valué *est un quintuplet* $(\mathscr{X}, \mathcal{D}, \mathscr{C}, \mathscr{V}, \varphi)$*, où :*
- ▷ $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ *est un réseau de contraintes classique ;*
- ▷ $\langle \mathscr{V}, \geq, \oplus \rangle$ *est une structure de valuation ;*
- ▷ φ *est une fonction de valuation fondée sur un ensemble de fonctions* $\{\varphi_c \mid c \in \mathscr{C}\}$ *telles que* $\varphi_c : Inst(\mathscr{X}(c)) \setminus \mathscr{R}(c) \to \mathscr{V}$

La notion de réseau de contraintes classique a été introduite dans la définition 3.3 en début de chapitre, et la structure de valuation est identique à celle introduite dans la définition 1.12 du chapitre 1. Le rôle des fonctions φ_c est d'associer à chaque instanciation violant la contrainte c un coût dans $\mathscr{V}$, représentant en quelque sorte la «gravité» du non-respect de la contrainte (l'élément $\top$ de la structure de valuation correspond à une instanciation interdite, donc à une contrainte qu'il faut absolument respecter). On voit donc ici que même si la structure de valuation est identique à celle introduite pour les structures de préférence, la sémantique qui lui est associée est ici toutefois légèrement différente, car on ne parle plus d'utilités (qu'il faut maximiser donc), mais de coûts de violation de contraintes (coût à minimiser). Cette légère différence est toutefois peu importante en pratique, et il suffit d'assimiler le coût à une «désutilité» associée à la violation d'une contrainte.

La valuation val d'une instanciation v est calculée par agrégation des valuations de toutes les contraintes violées par v :

$$val(v) = \bigoplus_{\substack{c \text{ tel que} \\ v_{\downarrow \mathscr{X}(c)} \in Inst(\mathscr{X}(c)) \setminus \mathscr{R}(c)}} \varphi_c(v_{\downarrow \mathscr{X}(c)}).$$

Bien entendu, on ne peut s'empêcher de remarquer la similitude formelle entre ce cadre et celui des utilités additives généralisées, bien que ces deux formalismes soient utilisés dans des contextes différents : satisfaction de contraintes pour les réseaux de contraintes valués (donc recherche de solutions minimisant le coût des contraintes violées), et élicitation ou agrégation de fonctions d'utilités pour les GAI-*nets*. Les réseaux de contraintes valués ont, tout comme les utilités GAI-décomposables, un formalisme graphique associé, fondé sur la notion de graphe de contraintes (ou hypergraphe de contraintes) et correspondant au graphe de dépendances entre variables dans les GAI-*nets*. De plus, comme nous l'avons fait remarquer, l'intérêt pour la décomposition arborescente (le calcul d'un arbre de jonction) est le même dans les graphes de contraintes que dans les GAI-*nets*.

Notons pour clore cette section sur les CSP valués que des travaux récents s'intéressent à la définition d'un cadre de modélisation et de résolution beaucoup plus générique que celui des CSP valués, qui intègre à la fois la notion de faisabilité d'une solution, la notion de plausibilité (faisant référence au concept de croyance), et enfin d'utilité. Le détail de ces travaux apparaît dans la thèse [Pralet, 2006].

3.2.5 Langages d'enchères combinatoires

Les enchères combinatoires sont un autre exemple de domaine d'étude récent dans lequel le besoin de représentation compacte de préférences est de première importance, ce qui le situe à l'interface entre l'économie, la représentation des connaissances, la complexité et l'algorithmique. Comme indiqué dans la description de ce problème (voir l'application 5 dans l'introduction), les enchères combinatoires se distinguent des enchères classiques par le fait que les acheteurs ont la possibilité de miser sur des lots d'objets, en plus de miser sur des objets indépendants. Cela transforme donc le problème initial, où les préférences des agents étaient représentés par des mises sur des objets individuels (donc au plus m valeurs à exprimer), en un problème par nature combinatoire, puisque l'espace des alternatives sur lequel les agents doivent miser (en d'autres termes exprimer leurs préférences) est un espace d'objets, donc de taille 2^m.

Le problème central des enchères combinatoires est le *Winner Determination Problem* :

Problème 1: Winner Determination Problem

INSTANCE :	Un ensemble d'agents $\mathscr{N}$, un ensemble d'objets $\mathscr{O}$, un ensemble de fonctions d'utilité $(f_1, \ldots, f_n)$ exprimées sous forme de mises dans un langage d'enchères combinatoires, et un entier K.
QUESTION :	Existe-t-il un partage $\overrightarrow{\pi}$ des objets qui satisfait la contrainte de préemption et tel que $\sum_{i=1}^{n} f_i(\pi_i) \geq K$?

Bien entendu, nous avons adapté la définition de ce problème au formalisme que nous avons introduit jusqu'ici, et on pourra trouver dans la littérature un ensemble de formulations légèrement différentes de celle-ci. Nous nous devons d'ajouter quelques précisions sur cette définition. Par «langage d'enchères combinatoires», nous entendons tout langage qui peut être utilisé par les agents pour exprimer des mises sur un ensemble de lots (autrement dit, tout langage numérique sur $\wp(\mathscr{O})$), tels que ceux que nous allons introduire dans cette sous-section. Comme nous allons le voir, les utilités individuelles représentent le prix que les agents sont prêts à payer pour un ensemble d'objets donné. Maximiser la somme des utilités individuelles revient donc à maximiser le revenu du commissaire-priseur (ce qui est un point de vue résolument utilitariste classique).

Nous allons présenter brièvement le paradigme dominant en matière de langages d'enchères combinatoires [Nisan, 2006]. Ces langages sont utilisés de manière explicite ou implicite dans la plupart des travaux portant sur le *Winner Determination Problem* [Sandholm, 1999; Fujishima *et al.*, 1999]. On pourra trouver une classification générale des langages d'enchères combinatoires dans [Nisan, 2000].

3.2.5.1 Langages OR et XOR

La plupart des langages permettant d'exprimer des préférences sur des espaces combinatoires d'objets que nous avons introduits ci-avant étaient fondés sur les objets comme «unité de base». Ainsi par exemple l'un des plus simples de ces langages, le langage additif, s'appuie sur l'ensemble des valuations attribués aux singletons d'objets. L'utilité d'un lot non réduit à un singleton se déduit ensuite par calcul à partir de ces valuations.

Dans le domaine des enchères combinatoires, le point de vue est légèrement différent puisque la plupart des langages que l'on peut trouver dans ce domaine sont centrés sur la notion de lot. En d'autres termes, ces langages sont fondés sur le principe que les acheteurs formulent un ensemble de mises[5] correspondant à des lots d'objets, mais n'ont que faire des objets individuels contenus

[5] Nous traduisons par «mise» le terme «*bid*» classique dans les enchères combinatoires, et par «lot» le terme

dans les lots. Un autre aspect des langages dédiés aux enchères combinatoires est que l'espace de valuation correspond en général à l'ensemble des prix que les agents associent aux lots d'objets : en d'autres termes, l'utilité d'un agent pour un lot donné est le prix que cet agent est disposé à payer pour ce lot[6]. Nous supposerons donc dans notre présentation des langages d'enchères combinatoires que l'espace de valuation est $\mathbb{R}^+$.

Le langage le plus basique des enchères combinatoires est le langage des mises atomiques : chaque agent n'a le droit de soumettre au commissaire-priseur qu'une seule mise, c'est-à-dire la donnée d'un seul couple (lot, valuation).

Définition 3.27 (Langage des mises atomiques) *Le langage de représentation de préférences $\mathcal{R}_{at.bids}$ est défini par :*
- ▷ $\mathcal{L}_{at.bids}(\mathscr{O}) = \wp(\mathscr{O}) \times \mathbb{R}^+$ *;*
- ▷ *pour toute mise (π, w), la structure de préférences cardinale induite $u_{at.bids}^{(\pi,w)}$ est définie par :*

$$\forall \pi' \in \wp(\mathscr{O}),\ u(\pi') = \begin{cases} w & si\ \pi' \supseteq \pi \\ 0 & sinon. \end{cases}$$

Le langage des mises atomiques, aussi appelé *single-minded bids language* dans [Lehmann *et al.*, 1999], est assez fruste et est très clairement incapable d'exprimer même une fonction d'utilité additive.

Dans les langages d'enchères combinatoires classiques, on ne limite plus l'expression des préférences à une seule mise atomique par agent, mais on suppose que chaque agent est capable d'en exprimer un nombre arbitraire. Les différentes manières d'interpréter cette multiplicité de mises atomiques conduisent à des langages différents. Le premier d'entre eux est le langage OR. Dans ce langage, la valeur d'une part est calculée en cherchant l'ensemble de lots mutuellement exclusifs contenus dans cette part, qui maximise la somme des prix associés.

Définition 3.28 (Langage OR) *Le langage de représentation de préférences $\mathcal{R}_{OR}$ est défini de la manière suivante :*
- ▷ *un élément de $\mathcal{L}_{OR}$ est un ensemble fini $\mathscr{M}_{OR}$ de mises atomiques, noté $(\pi_1, w_1)\ OR\ \ldots\ OR\ (\pi_p, w_p)$;*
- ▷ *pour tout ensemble de mises $\mathscr{M}_{OR} = (\pi_1, w_1)\ OR\ \ldots\ OR\ (\pi_p, w_p)$, la structure de préférences cardinale induite $u_{OR}^{\mathscr{M}_{OR}}$ est définie par :*

$$\forall \pi \in \wp(\mathscr{O}),\ u(\pi) = \max_{\mathscr{W}\ partition\ de\ \pi} \sum_{\pi' \in \mathscr{W}} w(\pi'),$$

$$avec\ w(\pi') = \begin{cases} w_k & si\ \exists (\pi_k, w_k) \in \mathscr{M}_{OR}\ tel\ que\ \pi' = \pi_k \\ 0 & sinon. \end{cases}$$

L'idée sous-jacente à la notion de mise OR est qu'un agent désire acquérir n'importe quel nombre de lots parmi les mises atomiques de la mise OR, et ce pour la somme des prix respectifs de ces mises. On remarquera qu'une mise OR correspond exactement à un ensemble de mises atomiques soumises par des enchérisseurs différents.

Ce langage de représentation de préférences est beaucoup plus expressif que celui des mises atomiques, puisqu'il est capable de représenter n'importe quelle fonction d'utilité u qui ne comporte pas de substituabilités (c'est-à-dire telle que $u(\pi \cup \pi') \geq u(\pi) + u(\pi')$).

«*bundle*».

[6]Remarquons toutefois que cet aspect n'est pas particulier aux enchères combinatoires, puisque de manière générale le prix qu'un agent est prêt à payer pour une alternative est un bon indicateur de son utilité pour cette alternative.

En revanche, ce gain d'expressivité se paie au prix fort, puisque le calcul de la valuation d'un lot quelconque est NP-complet. Cependant, ce saut de complexité n'est pas un très gros problème en pratique, car il est noyé dans le *Winner Determination Problem* global sans augmenter sa complexité.

Un autre langage est classique dans le domaine des enchères combinatoires : le langage XOR [Sandholm, 1999, 2002]. Contrairement au langage OR, il est complètement expressif (dans la limite de préférences monotones uniquement), et permet en particulier d'exprimer des substituabilités entre objets. Le langage XOR se distingue du langage OR en ceci que les lots sont tous mutuellement exclusifs, même s'ils ne possèdent aucun objet en commun. La signification sous-jacente à ce langage est qu'un agent ne désire qu'un et un seul lot parmi ceux de ses mises.

Définition 3.29 (Langage XOR) *Le langage de représentation de préférences $\mathcal{R}_{XOR}$ est défini de la manière suivante :*

- ▷ *un élément de $\mathcal{L}_{XOR}$ est un ensemble fini $\mathscr{M}_{XOR}$ de mises atomiques, noté (π_1, w_1) XOR … XOR (π_p, w_p) ;*
- ▷ *pour tout ensemble de mises $\mathscr{M}_{XOR} = (\pi_1, w_1)$ XOR … XOR (π_p, w_p), la structure de préférences cardinale induite $u_{XOR}^{\mathscr{M}_{XOR}}$ est définie par :*

$$\forall \pi \in \wp(\mathscr{O}), u(\pi) = \max_{\substack{(\pi_k, w_k) \in \mathscr{M}_{XOR} \\ \pi_k \subseteq \pi}} w_k.$$

Notons que si ce langage est pleinement expressif, en revanche il est incapable d'exprimer de manière compacte toutes les fonctions d'utilité représentables succinctement dans le langage OR. Ainsi, par exemple, une simple fonction d'utilité additive sur m objets, dont la représentation nécessite une mise OR de taille m, ne peut pas être représentée par une mise XOR de taille inférieure à 2^m.

3.2.5.2 Combiner les langages OR et XOR

La combinaison des langages OR et XOR est une extension naturelle des travaux sur la représentation compacte de mises. Cette combinaison permet d'allier l'expressivité du langage XOR à la compacité du langage OR. Le premier des langages fondé sur ce principe est le langage OR-of-XOR [Sandholm, 1999] :

Définition 3.30 (Langage OR-of-XOR) *Le langage de représentation de préférences $\mathcal{R}_{OoX}$ est défini de la manière suivante :*

- ▷ *un élément de $\mathcal{L}_{OoX}$ est un ensemble fini $\mathscr{M}_{OoX}$ de mises XOR : $\mathscr{M}_{OoX} = \mathscr{M}_{XOR}^1$ OR … OR $\mathscr{M}_{XOR}^p$, avec, pour tout k, $\mathscr{M}_{XOR}^k = (\pi_1^k, w(\pi_1^k))$ XOR … XOR $(\pi_{p_k}^k, w(\pi_{p_k}^k))$;*
- ▷ *pour tout ensemble de mises $\mathscr{M}_{OoX}$, la structure de préférences cardinale induite $u_{OoX}^{\mathscr{M}_{OoX}}$ est définie par :*

$$\forall \pi \in \wp(\mathscr{O}),\ u(\pi) = \max_{(\pi_1, \ldots, \pi_p) \in \mathcal{P}} \sum_{k=1}^{n} w(\pi_k),$$

avec $\mathcal{P}$ l'ensemble des séquences de lots $(\pi_1, \ldots, \pi_p)$ tels que
- $\forall i$, $\exists j$ *tel que* $\pi_i = \pi_j^i$ *(autrement dit π_i apparaît dans la $i^{\text{ème}}$ mise XOR) ou* $\pi_i = \varnothing$,
- $\forall i \neq j$, $\pi_i \cap \pi_j = \varnothing$.

En d'autres termes, un enchérisseur qui exprime une mise OR-of-XOR ne désire qu'au plus un lot par mise XOR, les lots sélectionnés pour chaque mise XOR étant mutuellement exclusifs. Le langage OR-of-XOR généralise à la fois le langage OR et le langage XOR.

Exemple 3.12 L'agent concerné par le problème de l'héritage (problème 3.2) peut exprimer de manière naturelle ses préférences sous la forme OR-of-XOR. Il peut exprimer notamment le fait qu'il ne veuille qu'un seul bien immobilier, et qu'un seul bien parmi la tondeuse et le tracteur : $((\{ch\}, 5\ 000\ €)\ XOR\ (\{m\}, 60\ 000\ €))\ OR\ ((\{to\}, 500\ €)\ XOR\ (\{tr\}, 3\ 000\ €))$. Par exemple, si l'agent obtient les terres cultivables (ch) et le tracteur (tr), il évaluera ce lot à 8 000 €. S'il obtient en plus la maison, le lot sera évalué à 63 000 € (dans ce cas, les terres cultivables ne comptent plus car elles sont éliminées par la maison dans la mise XOR correspondante).

Si l'agent voulait exprimer ces mêmes préférences dans le langage XOR, il lui faudrait une mise de taille 8 :

$$\begin{array}{cccccccc}
(\{ch\}, 5\ 000\ €) & XOR & (\{m\}, 60\ 000\ €) & XOR & (\{to\}, 500\ €) & XOR \\
(\{tr\}, 3\ 000\ €) & XOR & (\{ch, to\}, 5\ 500\ €) & XOR & (\{ch, tr\}, 8\ 000\ €) & XOR \\
(\{m, to\}, 60\ 500\ €) & XOR & (\{m, tr\}, 63\ 000\ €). & & &
\end{array}$$

La combinaison duale des langages OR et XOR existe aussi, sous la forme du langage XOR-of-OR. Bien que moins utilisé dans le domaine des enchères combinatoires, en raison de son aspect légèrement moins intuitif que le langage OR-of-XOR, notons tout de même qu'il permet de représenter de manière plus succincte que ce dernier langage un ensemble de fonctions d'utilité telles que les fonctions monochromatiques (de manière informelle, les fonctions d'utilité telles que, chaque objet ayant une couleur bien définie, l'agent ne désire que des objets de couleur identique).

Nous citerons enfin deux autres extensions des langages fondés sur des combinaisons des langages OR et XOR. Le langage OR/XOR est fondé sur une combinaison arbitraire de mises formées à partir de ces deux opérateurs. Formellement, une formule du langage OR/XOR est une mise atomique ou formule du type $\mathscr{M}_1\ OR\ \mathscr{M}_2$ ou $\mathscr{M}_1\ XOR\ \mathscr{M}_2$, avec $\mathscr{M}_1$ et $\mathscr{M}_2$ des formules du langage OR/XOR. Les langages OR, XOR, OR-of-XOR et XOR-of-OR sont des cas particuliers de ce langage générique.

Un dernier langage d'importance dans le domaine des enchères combinatoires [Fujishima *et al.*, 1999; Nisan, 2000] est le langage OR*. Ce langage est fondé sur le langage OR, avec lequel il partage la même syntaxe, mais il permet de simuler des mises mutuellement exclusives par l'introduction d'objets factices. Le rôle de ces objets est simplement d'empêcher l'attribution simultanée de deux lots, et ainsi de simuler des mises XOR. Par exemple, si un agent désire placer la mise $(\pi_1, w_1)\ XOR\ (\pi_2, w_2)$, il peut introduire un objet factice d («d» pour *dummy*) et exprimer la mise précédente par une mise OR : $(\pi_1 \cup \{d\}, w_1)\ OR\ (\pi_2 \cup \{d\}, w_2)$. L'intérêt principal de ce langage, et la raison de son succès dans le domaine des enchères combinatoires est qu'il permet de simuler des mises du type XOR, tout en gardant une syntaxe identique à celle du langage OR, permettant ainsi à une instance du *Winner Determination Problem* exprimée dans le langage OR* d'être traitée par un solveur dédié aux instances exprimées dans le langage OR. Notons que ce langage est celui qui est utilisé par le générateur CATS [Leyton-Brown *et al.*, 2000], devenu une référence en terme de génération d'instances du problème d'enchères combinatoires.

3.2.5.3 Langages logiques pour les enchères combinatoires

Si le paradigme dominant dans le domaine de la représentation de préférences dans les enchères combinatoires est fondé sur les langages OR et XOR et leurs combinaisons, d'autres travaux récents du domaine s'appuient sur des langages à base de logique propositionnelle pondérée [Boutilier et Hoos, 2000, 2001]. Ces langages logiques ressemblent à la fois aux langages que nous avons introduits plus tôt dans ce chapitre s'appuyant sur la logique propositionnelle et aux langages de lots introduits dans cette section. En effet, ils permettent d'exprimer à la fois des combinaisons logiques pondérées d'objets simples et des combinaisons logiques de lots, cumulant les avantages de ces deux approches.

Décrivons rapidement le langage logique introduit et décrit en détails dans [Boutilier et Hoos, 2001]. Ce langage étend le langage logique à base de buts pondérés introduit dans la définition 3.18 en ceci qu'il permet d'attribuer des valuations à des sous-formules, et non uniquement à des formules entières. En d'autres termes, si, dans le langage à base de buts pondérés, les préférences d'un agent étaient représentées par un ensemble de formules logiques pondérées $(\varphi_1, w_1), \ldots, (\varphi_p, w_p)$, dans ce nouveau langage, les préférences d'un agent sont représentées par une formule d'un langage $\mathcal{L}_{GWL}$ («*GWL*» pour *Generalized Weighted Logic*) dont la syntaxe est la suivante :

- ▷ pour tout $o \in \mathscr{O}$ et $w \in \mathbb{R}^+$, $\langle o, w \rangle$ est une formule de $\mathcal{L}_{GWL}$;
- ▷ pour toutes formules b_1 et b_2 de $\mathcal{L}_{GWL}$ et tout $w \in \mathbb{R}^+$, $\langle b_1 \wedge b_2, w \rangle$ et $\langle b_1 \vee b_2, w \rangle$ sont des formules de $\mathcal{L}_{GWL}$ ([Boutilier et Hoos, 2001] introduit de plus un opérateur logique ou-exclusif que nous n'introduisons pas ici pour simplifier).

L'utilité d'un lot est calculée par agrégation des poids de toutes les sous-formules satisfaites. Ainsi, par exemple et de manière informelle, l'utilité d'un lot π vis-à-vis d'une formule du type $\langle b_1 \wedge b_2, w \rangle$ est calculée en agrégeant :

- ▷ l'utilité de π vis-à-vis de b_1 ;
- ▷ l'utilité de π vis-à-vis de b_2 ;
- ▷ w si π «satisfait» $b_1 \wedge b_2$.

Comme nous l'avons fait remarquer ci-avant, ce langage cumule les avantages de l'approche fondée sur les objets et de l'approche fondée sur les lots, en autorisant la combinaison logique de formules pondérées (qui peuvent s'apparenter à des mises). Cette expressivité a probablement une contrepartie en terme de difficulté de résolution du *Winner Determination Problem*, bien que cette difficulté ne semble pas encore avoir été réellement mise en évidence de manière empirique. L'article [Boutilier et Hoos, 2001] s'oriente en tout cas vers un algorithme de résolution fondé sur la recherche locale stochastique, plutôt que sur une approche exacte comme dans l'approche traditionnelle des enchères combinatoires.

3.2.6 Conclusion sur les langages de représentation de préférences

Nous avons tenté de présenter une vue d'ensemble de l'ensemble des langages d'expression compacte classiques de la littérature sur la représentation des préférences. Nous avons distingué plusieurs types de langages :

- ▷ les langages à base de logique, très expressifs et intuitifs, et relativement adaptés dans le cadre du partage ;
- ▷ les langages à base de préférences *ceteris paribus*, qui offrent une alternative aux langages logiques relativement élégante et intéressante d'un point de vue cognitif, mais qui ne sont pas toujours très pertinents dans le cadre des problèmes de partage ;
- ▷ les langages fondés sur l'indépendance additive généralisée, dont les langages de lots k-additifs sont étudiés dans le cadre du partage ;
- ▷ les langages d'enchères combinatoires, centrés sur l'expression de mises sur des lots.

Nous avons résumé l'ensemble de ces langages dans le tableau 3.1.

3.3 Représentation compacte des problèmes de partage équitable

Nous avons jusqu'ici dressé un aperçu des principaux langages de représentation compacte d'espace combinatoires et de préférences sur ces espaces, et nous avons brièvement discuté de la pertinence de ces langages dans le cadre des problèmes de partage. Nous allons dans cette section nous appuyer sur ces langages pour définir deux langages de représentation compacte de problèmes de partage équitable.

Langage	syntaxe	type	var. binaires
Langages logiques			
Dichotomique	$\mathcal{L}_{\mathcal{O}}$	dichotomique	oui
Pareto	$\mathcal{GB}_{\mathcal{O}}$	ordinale	oui
Leximin, discrimin, *best-out*	$\mathcal{GB}_{\mathcal{O}}^{strat}$	ordinale	oui
Card	$\mathcal{GB}_{\mathcal{O}}$	cardinale	oui
Distances	$\mathcal{GB}_{\mathcal{O}}$	cardinale	oui
Logique pondérée	$\mathcal{GB}_{\mathcal{O}}^{weighted}$	cardinale	oui
Préférences *ceteris paribus*			
Ceteris Paribus	relations entre formules $\lambda : \varphi \rhd \psi$	ordinale	oui
CP-*nets*	Graphe dirigé, tables de préférences CP	ordinale	non
TCP-*nets*	Graphe dirigé possédant 3 types d'arcs, tables de préférences CP	ordinale	non
UCP-*nets*	Graphe dirigé, tables de préférences CP avec des utilités	cardinale	non
Additivité généralisée			
Additif	Utilités $u(o)$	cardinale	oui
k-additif	Utilités $u(\pi)$, pour $\|\pi\| \leq k$	cardinale	oui
GAI-*net*	Graphe, tables de valuations GAI	cardinale	non
CSP valué	Spécification de contraintes valuées	cardinale	non
Langages de lots pour les enchères combinatoires			
OR	$(\pi_1, w_1)\ OR\ (\pi_2, w_2)$	cardinale	oui
XOR	$(\pi_1, w_1)\ XOR\ (\pi_2, w_2)$	cardinale	oui
OR-of-XOR, XOR-of-OR, XOR/OR	Combinaisons de mises OR ou XOR	cardinale	oui
Logique pondérée généralisée	Combinaisons logiques pondérées de sous-formules logiques pondérées	cardinale	oui

Tableau 3.1 — Liste des langages de représentation de préférences introduits dans ce chapitre.

Comme nous l'avons signalé dans l'introduction du manuscrit, les problèmes de partage équitable n'ont été que rarement — et très récemment — étudiés sous l'angle de la représentation compacte (et de la complexité). Le besoin de représentation compacte dans les problèmes de partage naît du dilemme suivant, qui a été formulé par plusieurs théoriciens du choix social. (a) Soit on autorise les agents à exprimer n'importe quelle relation de préférence sur l'ensemble de tous les sous-ensembles d'objets, et l'on a à faire face à une représentation exponentiellement large. C'est le point de vue de [Herreiner et Puppe, 2002], qui suppose que les agents ont des préférences linéaires (un ordre strict) sur l'ensemble $\wp(\mathscr{O})$, et propose des procédures relativement intéressantes pour traiter avec ce genre de préférences. Cependant, comme nous l'avons vu dans l'introduction, l'explosion combinatoire liée à l'expression des préférences rend de telles procédures inutilisables en pratique. (b) La deuxième solution est de limiter de manière drastique l'ensemble des préférences exprimables, en supposant typiquement l'indépendance additive entre les objets. C'est la voie suivie par [Brams *et al.*, 2003] et [Demko et Hill, 1998]. Cependant, comme nous l'avons vu dans ce chapitre, il est possible de concilier les deux approches : c'est l'objectif de la représentation compacte.

Nous allons nous intéresser à la représentation compacte de deux problèmes de partage équitable particuliers, correspondant aux deux visions de l'équité présentées dans le chapitre 1. Le premier de ces problèmes est lié à la propriété d'absence d'envie. L'absence d'envie est une propriété très intéressante, mais n'est cependant pas suffisante pour assurer la qualité du partage : il faut un critère d'efficacité. Nous nous intéresserons donc en particulier au problème de recherche d'un partage Pareto-efficace et sans envie. Nous aborderons ce problème sous l'angle des préférences les plus simples qui soient, c'est-à-dire les préférences dichotomiques représentées sous forme logique, et en présence de la contrainte de préemption uniquement. Nous verrons que dans ce cadre, l'absence d'envie et la Pareto-efficacité s'expriment comme des propriétés logiques simples.

Le second problème auquel nous allons nous intéresser est le problème de maximisation d'une fonction d'utilité collective, lorsque les agents ont des préférences numériques exprimées sous forme logique. Nous fournirons un cadre de représentation compacte générique pour ce problème, cadre dont l'intérêt sera d'être assez expressif pour représenter un large spectre de problèmes de partage de biens indivisibles.

3.3.1 Pareto-efficacité et absence d'envie en présence de préférences dichotomiques : représentation logique

Nous allons donc nous intéresser dans un premier temps à la structure de préférences dichotomique (voir définition 1.10). Comme nous l'avons vu, une structure de préférences dichotomique est une structure de préférences ordinale dégénérée, pour laquelle il n'existe que deux niveaux de satisfaction pour les alternatives : un ensemble de «bonnes» alternatives (noté $\mathscr{G}$), et un ensemble de «mauvaises» alternatives. Toute bonne alternative est strictement meilleure qu'une mauvaise alternative ; mais les bonnes alternatives sont indifférentes entre elles, de même que les mauvaises entre elles.

Comme nous l'avons vu dans la section 3.2.2.1, il existe une manière évidente d'exprimer une structure de préférences dichotomique de manière compacte sur un ensemble de parts d'un ensemble d'objets $\mathscr{O}$: il suffit d'introduire une formule propositionnelle φ sur le langage $\mathcal{L}_{\mathscr{O}}$, tel que $Mod(\varphi) = \mathscr{G}$.

Exemple 3.13 $\mathscr{O} = \{o_1, o_2, o_3\}$ et $\mathscr{G} = \{\{o_1, o_2\}, \{o_2, o_3\}\}$. On peut remarquer que la relation de préférences $\succeq$ correspondant à cet ensemble $\mathscr{G}$ n'est pas monotone. Alors $\varphi = (\mathbf{o_1} \wedge \mathbf{o_2} \wedge \neg\mathbf{o_3}) \vee (\neg\mathbf{o_1} \wedge \mathbf{o_2} \wedge \mathbf{o_3})$ représente $\succeq$, de même que $\varphi' = \mathbf{o_2} \wedge ((\mathbf{o_1} \wedge \neg\mathbf{o_3}) \vee (\neg\mathbf{o_1} \wedge \mathbf{o_3}))$, qui est logiquement équivalente à φ.

On peut légitimement se demander quel intérêt on peut porter à un langage de représentation de préférences aussi frustre. Tout d'abord, l'hypothèse de dichotomie des préférences est raisonnable dans un certain nombre de problèmes, comme par exemple les problèmes pour lesquels les agents ne désirent qu'un seul objet ou ensemble d'objets. Cependant, l'intérêt principal de cette structure de préférences est «computationnel». Sa relative simplicité permet d'exprimer la plupart des propriétés ayant trait au partage comme des propriétés logiques, et de se ramener ainsi à des grands problèmes classiques de la logique propositionnelle, comme nous allons le voir. Cependant, malgré la simplicité de cette structure de préférences et de son langage de représentation compacte associé, nous allons voir que ce langage concentre toute la complexité du problème de partage liée à la représentation compacte, à l'efficacité et à l'absence d'envie : en d'autres termes, il n'est pas plus facile de résoudre un problème de partage avec des préférences dichotomiques qu'avec des préférences plus générales munies de leur langage de représentation compacte raisonnable. La structure de préférences dichotomique et son langage logique associé constituent donc un point d'entrée idéal pour l'étude de la complexité de l'existence d'un problème de partage efficace et sans envie.

Introduisons un résultat facile à obtenir mais utile pour la suite :

Proposition 3.3 *Soit $\succeq_i$ une structure de préférences dichotomique sur $\wp(\mathscr{O})$. Alors les affirmations suivantes sont équivalentes :*

1. *$\succeq_i$ est monotone ;*
2. *$\mathscr{G}_i$ est supérieurement clos, c'est-à-dire que $\mathscr{S} \in \mathscr{G}_i$ et $\mathscr{S}' \supseteq \mathscr{S}$ implique que $\mathscr{S}' \in \mathscr{G}_i$.*
3. *$\succeq_i$ peut être représentée dans $\mathcal{R}_{dicho}$ par une formule propositionnelle positive.*

Démonstration (1) $\Rightarrow$ (2). Supposons que $\succeq_i$ est monotone ; soient $\mathscr{S} \in \mathscr{G}_i$ et $\mathscr{S}' \supseteq \mathscr{S}$. Alors nous avons forcément $\mathscr{S}' \succeq_i \mathscr{S}$, et donc $\mathscr{S}' \in \mathscr{G}_i$ (puisque $\mathscr{S} \in \mathscr{G}_i$).

(2) $\Rightarrow$ (3). Supposons que $\mathscr{G}_i$ est supérieurement clos, et considérons l'ensemble $\min_\subseteq(\mathscr{G}_i)$ de tous les ensembles de $\mathscr{G}_i$ minimaux pour l'inclusion. Alors la formule $\varphi_i = \bigvee_{\mathscr{S} \in \min_\subseteq(\mathscr{G}_i)} \left(\bigwedge_{o \in \mathscr{S}} \mathbf{o}\right)$ représente $\succeq_i$ pour les raisons suivantes. Pour tout $\mathscr{S}' \in \mathscr{G}_i$, il existe un ensemble $\mathscr{S} \in \min_\subseteq(\mathscr{G}_i)$ tel que $\mathscr{S} \subseteq \mathscr{S}'$. Donc la conjonction correspondante dans φ_i est satisfaite, et donc φ_i est satisfaite. Réciproquement, pour tout ensemble $\mathscr{S}' \notin \mathscr{G}_i$, il n'existe aucun $\mathscr{S} \subseteq \min_\subseteq(\mathscr{G}_i)$ tel que $\mathscr{S} \in \mathscr{S}'$. En conséquence, aucun des cubes de φ_i n'est satisfait, et donc la formule φ_i est insatisfaite. De plus, φ_i est clairement une formule propositionnelle positive.

(3) $\Rightarrow$ (1). Supposons que $\succeq_i$ peut être représentée par une formule propositionnelle positive φ_i, et soient $\mathscr{S}$ et $\mathscr{S}'$ deux ensembles d'objets tels que $\mathscr{S} \subseteq \mathscr{S}'$. Si $\mathscr{S} \notin \mathscr{G}_i$, alors très clairement $\mathscr{S}' \succeq_i \mathscr{S}$. Si $\mathscr{S} \in \mathscr{G}_i$, alors $\mathscr{S} \in Mod(\varphi_i)$. Puisque φ_i est positive, alors on a aussi $\mathscr{S}' \in Mod(\varphi_i)$. En conséquence, $\mathscr{S}' \in \mathscr{G}_i$, et donc au final $\mathscr{S}' \succeq_i \mathscr{S}$. ▲

La donnée des agents, des objets, et des formules logiques représentant les préférences des agents suffit donc à définir une instance du problème de partage équitable de biens indivisibles avec préférences dichotomiques. Nous supposerons donc à partir de maintenant que les instances $\mathcal{P}$ de ce problème sont représentées sous la forme $(\mathscr{N}, \mathscr{O}, (\varphi_1, \ldots, \varphi_n))$ au lieu de spécifier $\mathscr{N}$, $\mathscr{O}$ et l'ensemble des relations de préférence $(\succeq_1, \ldots, \succeq_n)$.

Soit $\mathcal{P} = (\mathscr{N}, \mathscr{O}, (\varphi_1, \ldots, \varphi_n))$ un problème d'allocation avec préférences dichotomiques. Les formules φ_i impliquent toutes les mêmes variables propositionnelles sur $\mathscr{O}$. Dans la suite de la construction du modèle, nous allons avoir besoin de différentier les variables entre les formules. Nous allons donc transposer les préférences des agents dans le langage $\mathcal{L}_{\mathscr{O}}^{alloc}$. Plus concrètement, pour chaque $i \in \mathscr{N}$, nous notons φ_i^* la formule φ_i dans laquelle on a remplacé chaque variable $\mathbf{o}$ par le symbole $\mathbf{alloc}(\mathbf{o}, \mathbf{i})$.

Exemple 3.14 Considérons le problème de partage suivant : $\mathscr{N} = \{1, 2, 3\}$, $\mathscr{O} = \{o_1, o_2, o_3\}$, $\varphi_1 = \mathbf{o_1} \vee (\mathbf{o_2} \wedge \mathbf{o_3})$, $\varphi_2 = \mathbf{o_1}$ et $\varphi_3 = \mathbf{o_1} \vee \mathbf{o_2}$. Toutes les formules sont positives, et donc les préférences sont monotones. Par exemple, $\mathscr{G}_1$ est l'ensemble composé de tous les sur-ensembles de $\{\mathbf{o_1}\}$ et tous les sur-ensembles de $\{\mathbf{o_2}, \mathbf{o_3}\}$. Nous avons alors :

$$\begin{aligned} \varphi_1^* &= \mathbf{alloc(o_1, 1)} \vee (\mathbf{alloc(o_2, 1)} \wedge \mathbf{alloc(o_3, 1)}); \\ \varphi_2^* &= \mathbf{alloc(o_2, 2)}; \\ \varphi_3^* &= \mathbf{alloc(o_1, 3)} \vee \mathbf{alloc(o_2, 3)}. \end{aligned}$$

Comme nous l'avons précisé en tout début de section, nous nous limitons pour ce modèle aux problèmes de partage sans autre contrainte que la contrainte de préemption. Cette contrainte peut bien entendu être représentée sous forme logique, à l'aide de la formule suivante : $\Gamma_{\mathcal{P}} = \bigwedge_{o \in \mathscr{O}} \bigwedge_{i \neq j} \neg(\mathbf{alloc(o, i)} \wedge \mathbf{alloc(o, j)})$. En d'autres termes, les seuls partages admissibles sont ceux qui correspondent à un modèle de $Alloc_{\mathscr{O},\mathscr{N}}$ qui satisfait *au plus* un des $\mathbf{alloc(o, i)}$ pour tout $o \in \mathscr{O}$. Nous pouvons donc associer à toute interprétation $M \in Mod(\Gamma_{\mathcal{P}})$ un partage $\overrightarrow{\pi}$ défini simplement par $\pi_i = \{\mathbf{x} \mid M \models \mathbf{x_i}\}$. Cette transformation est clairement bijective, et donc nous pouvons associer à tout partage $\overrightarrow{\pi}$ une interprétation de $Mod(\Gamma_{\mathcal{P}})$, que nous noterons $F(\overrightarrow{\pi})$.

Exemple 3.14.a Reprenons le problème de partage présenté dans l'exemple 3.14. Par souci de place, nous abrégerons, dans cet exemple et dans le suivant uniquement, $\mathbf{alloc(o_i, j)}$ en $\mathbf{o_i^j}$. Nous avons :

$$\begin{aligned} \Gamma_{\mathcal{P}} = \quad & \neg(\mathbf{o_1^1} \wedge \mathbf{o_1^2}) \wedge \neg(\mathbf{o_1^1} \wedge \mathbf{o_1^3}) \wedge \neg(\mathbf{o_1^2} \wedge \mathbf{o_1^3}) \\ \wedge \quad & \neg(\mathbf{o_2^1} \wedge \mathbf{o_2^2}) \wedge \neg(\mathbf{o_2^1} \wedge \mathbf{o_2^3}) \wedge \neg(\mathbf{o_2^2} \wedge \mathbf{o_2^3}) \\ \wedge \quad & \neg(\mathbf{o_3^1} \wedge \mathbf{o_3^2}) \wedge \neg(\mathbf{o_3^1} \wedge \mathbf{o_3^3}) \wedge \neg(\mathbf{o_3^2} \wedge \mathbf{o_3^3}). \end{aligned}$$

L'interprétation M telle que M n'instancie que $\mathbf{o_1^1}$, $\mathbf{o_2^3}$ et $\mathbf{o_3^1}$ à vrai est très clairement un modèle de $\Gamma_{\mathcal{P}}$. Ce modèle correspond à l'allocation $F^{-1}(M) = \overrightarrow{\pi}$, avec $\pi_1 = \{\mathbf{o_1}, \mathbf{o_3}\}$, $\pi_2 = \varnothing$ et $\pi_3 = \{\mathbf{o_2}\}$.

3.3.1.1 Absence d'envie

Nous allons maintenant montrer comment le problème de recherche d'un partage sans envie peut être transformé en un problème de recherche de modèle dans une formule propositionnelle. Soit $\varphi_{j|i}^*$ la formule obtenue à partir de φ_i^* en substituant chaque symbole $\mathbf{alloc(o, i)}$ dans φ_i^* par le symbole $\mathbf{alloc(o, j)}$: par exemple, si $\varphi_1^* = \mathbf{alloc(o_1, 1)} \wedge (\mathbf{alloc(o_2, 1)} \vee \mathbf{alloc(o_3, 1)})$ alors $\varphi_{2|1}^* = \mathbf{alloc(o1, 2)} \wedge (\mathbf{alloc(o_2, 2)} \vee \mathbf{alloc(o_3, 2)})$. Remarquons que l'on a bien entendu $\varphi_{i|i}^* = \varphi_i^*$.

Nous introduisons le lemme suivant, qui est évident mais très utile :

Lemme 1 *Pour tout (i, j), $\pi_j \in \mathscr{G}_i$ si et seulement si $F(\overrightarrow{\pi}) \models \varphi_{j|i}^*$.*

En particulier, lorsque $i = j$ on a $\pi_i \in \mathscr{G}_i$ si et seulement si $F(\overrightarrow{\pi}) \models \varphi_i^*$.

> **Démonstration** Par définition de F, $\pi_j \in \mathscr{G}_i$ si et seulement si $\{o \mid F(\overrightarrow{\pi}) \models \mathbf{alloc(o, j)}\} \in \mathscr{G}_i$, c'est-à-dire $\{o \mid F(\overrightarrow{\pi}) \models \mathbf{alloc(o, j)}\} \models \varphi_i$. Cette dernière relation est équivalente à $\{\mathbf{alloc(o, i)} \mid F(\overrightarrow{\pi}) \models \mathbf{alloc(o, j)}\} \models \varphi_i^*$, et enfin à $\{\mathbf{alloc(o, j)} \mid F(\overrightarrow{\pi}) \models \mathbf{alloc(o, j)}\} \models \varphi_{j|i}^*$ par définition de $\varphi_{j|i}^*$. Nous pouvons donc en déduire le résultat. ▲

À l'aide de ce lemme, nous pouvons maintenant exprimer la propriété d'absence d'envie comme une propriété de satisfiabilité d'une formule logique :

Proposition 3.4 *Soit $\mathcal{P} = (\mathscr{N}, \mathscr{O}, (\varphi_1, \ldots, \varphi_n))$ une instance du problème de partage avec préférences dichotomiques sous forme propositionnelle, et soient $\varphi^*_{j|i}$ la formule et F la bijection définis ci-avant. Soit :*

$$\Lambda_{\mathcal{P}} = \bigwedge_{i=1,\ldots,n} \left[\varphi^*_i \vee \left(\bigwedge_{j \neq i} \neg \varphi^*_{j|i} \right) \right].$$

Alors $\overrightarrow{\pi}$ est sans envie si et seulement si $F(\overrightarrow{\pi}) \models \Lambda_{\mathcal{P}}$.

Démonstration Soit $\overrightarrow{\pi}$ une allocation. $\overrightarrow{\pi}$ n'est pas sans envie si et seulement s'il existe une paire d'agents (i, j), avec $i \neq j$, telle que $\pi_j \succ_i \pi_i$, c'est-à-dire $\pi_j \in \mathscr{G}_i$ et $\pi_i \notin \mathscr{G}_i$, ce qui est à son tour équivalent à $F(\overrightarrow{\pi}) \models \varphi^*_{j|i}$ et $F(\overrightarrow{\pi}) \not\models \varphi^*_i$ d'après le lemme 1. Donc π est sans envie si et seulement si $F(\overrightarrow{\pi}) \models \Lambda_{\mathcal{P}}$. ▲

La recherche d'allocations sans envie peut donc se ramener à un problème de recherche de modèles : $\{F^{-1}(M) \mid M \models \Gamma_{\mathcal{P}} \wedge \Lambda_{\mathcal{P}}\}$ est l'ensemble des allocations sans envie pour $\mathcal{P}$. Remarquons bien que $\Gamma_{\mathcal{P}} \wedge \Lambda_{\mathcal{P}}$ a une taille polynomiale (précisément, quadratique) en la taille des données d'entrée. Cette remarque est importante pour les résultats de complexité du chapitre 4.

Comme nous l'avons déjà fait remarquer, rechercher un partage sans envie (sans requérir aucune autre propriété sur le partage) n'est pas très intéressant, car une telle allocation existe toujours : il suffit de considérer le partage qui donne une part vide à tous les agents. Cependant :

- ▷ le problème de déterminer s'il existe une allocation sans envie vérifiant une propriété donnée exprimable par une formule logique de taille polynomiale (par exemple la propriété de complétude du partage) peut se réduire à un problème de satisfiabilité ;
- ▷ le problème de recherche (resp. de décompte) de tous les partages sans envie se ramène à un problème de recherche (resp. de décompte) de tous les modèles de la formule $\Lambda_{\mathcal{P}} \wedge \Gamma_{\mathcal{P}}$.

Exemple 3.14.b Revenons sur le problème de l'exemple 3.14. Nous avons :

$$\begin{array}{rl} \Lambda_{\mathcal{P}} = & [(\mathbf{o^1_1} \vee (\mathbf{o^1_2} \wedge \mathbf{o^1_3})) \vee (\neg(\mathbf{o^2_1} \vee (\mathbf{o^2_2} \wedge \mathbf{o^2_3})) \wedge \neg(\mathbf{o^3_1} \vee (\mathbf{o^3_2} \wedge \mathbf{o^3_3})))] \\ \wedge & [\mathbf{o^2_1} \vee (\neg \mathbf{o^1_1} \wedge \neg \mathbf{o^3_1})] \\ \wedge & [(\mathbf{o^3_1} \vee \mathbf{o^3_2}) \vee (\neg(\mathbf{o^1_1} \vee \mathbf{o^1_2}) \wedge \neg(\mathbf{o^2_1} \vee \mathbf{o^2_2}))] \end{array}$$

$$Mod(\Gamma_{\mathcal{P}} \wedge \Lambda_{\mathcal{P}}) = \{\{\mathbf{o^1_3}\}, \{\mathbf{o^1_3}, \mathbf{o^3_2}\}, \{\mathbf{o^2_3}, \mathbf{o^3_2}\}, \{\mathbf{o^2_3}\}, \{\mathbf{o^3_2}\}, \{\mathbf{o^3_3}\}, \varnothing\}.$$

Il y a donc 7 allocations sans envie, qui sont les suivantes : $(o_3, -, -)$, $(o_3, -, o_2)$, $(-, o_3, o_2)$, $(-, o_3, -)$, $(-, -, o_2)$, $(-, -, o_3)$ et $(-, -, -)$. Notons qu'aucune d'entre elles n'est sans envie.

3.3.1.2 Partages efficaces

De manière similaire à la propriété d'absence d'envie, la propriété de Pareto-efficacité peut être traduite en une propriété logique. L'expression logique de cette propriété requiert la définition de la notion de sous-ensemble maximal β-consistant d'un ensemble de variables.

Définition 3.31 *Soient $\Delta = \{\alpha_1, \ldots, \alpha_m\}$ un ensemble de formules et β une formule. $\mathscr{S} \subseteq \Delta$ est un* sous-ensemble maximal β-consistant de Δ *si et seulement si (a) $\bigwedge \mathscr{S} \wedge \beta$ est consistant et (b) il n'existe aucun ensemble $\mathscr{S}'$ tel que $\mathscr{S} \subset \mathscr{S}' \subseteq \Delta$ et $\bigwedge \mathscr{S}' \wedge \beta$ est consistant (satisfiable). Soit $MaxCons(\Delta, \beta)$ l'ensemble de tous les sous-ensembles maximaux β-consistants de Δ. Nous noterons de plus $MaxCons(\Delta)$ l'ensemble de tous les sous-ensembles maximaux-consistants de Δ, c'est-à-dire l'ensemble $MaxCons(\Delta, \top)$.*

Proposition 3.5 *Soit $\mathcal{P} = (\mathscr{N}, \mathscr{O}, (\varphi_1, \dots, \varphi_n))$ un problème de partage. Soit $\Phi_\mathcal{P} = \{\varphi_1^*, \dots, \varphi_n^*\}$. Alors $\overrightarrow{\pi}$ est Pareto-efficace pour $\mathcal{P}$ si et seulement si $\{\varphi_i^* \mid F(\overrightarrow{\pi}) \vDash \varphi_i^*\}$ est un sous-ensemble maximal $\Gamma_\mathcal{P}$-consistant de $\Phi_\mathcal{P}$.*

Démonstration Soit $\overrightarrow{\pi}$ une allocation. Soit $Sat(\overrightarrow{\pi})$ l'ensemble des agents satisfaits par $\overrightarrow{\pi}$, c'est-à-dire, d'après le lemme 1, l'ensemble $\{i \mid F(\overrightarrow{\pi}) \vDash \varphi_i^*\}$. Pour tout $i \in Sat(\overrightarrow{\pi})$, $F(\overrightarrow{\pi}) \vDash \varphi_i^*$ par définition de $Sat(\overrightarrow{\pi})$, et $F(\overrightarrow{\pi}) \vDash \Gamma_\mathcal{P}$ par définition de $F(\overrightarrow{\pi})$. Donc $F(\overrightarrow{\pi}) \vDash \Gamma_\mathcal{P} \wedge \bigwedge\{\varphi_i^* \mid i \in Sat(\overrightarrow{\pi})\} = \Gamma_\mathcal{P} \wedge \bigwedge\{\varphi_i^* \mid F(\overrightarrow{\pi}) \vDash \varphi_i^*\}$. En conséquence $\bigwedge\{\varphi_i^* \mid F(\overrightarrow{\pi}) \vDash \varphi_i^*\} \wedge \Gamma_\mathcal{P}$ est consistant.

Par définition, $\overrightarrow{\pi}$ est Pareto-dominé si et seulement s'il existe une allocation $\overrightarrow{\pi}'$ telle que $Sat(\overrightarrow{\pi}') \supsetneq Sat(\overrightarrow{\pi})$. En conséquence, si $\overrightarrow{\pi}$ est Pareto-dominé, il existe un sous-ensemble consistant $\mathscr{S} \subseteq \Phi_\mathcal{P}$ (correspondant à $\{\varphi_i^* \mid F(\overrightarrow{\pi}') \vDash \varphi_i^*\}$) tel que $\{\varphi_i^* \mid F(\overrightarrow{\pi}) \vDash \varphi_i^*\} \subset \mathscr{S}$. De plus, puisque $\overrightarrow{\pi}'$ est un partage, $\bigwedge \mathscr{S} \wedge \Gamma_\mathcal{P}$ est consistant.

Réciproquement, soit $\mathscr{S} \subseteq \Phi_\mathcal{P}$ tel que $\{\varphi_i^* \mid F(\overrightarrow{\pi}) \vDash \varphi_i^*\} \subset \mathscr{S}$ et $\bigwedge \mathscr{S} \wedge \Gamma_\mathcal{P}$ est consistant, et soit M un modèle de $\bigwedge \mathscr{S} \wedge \Gamma_\mathcal{P}$. Par définition, $\overrightarrow{\pi}' = F^{-1}(M)$ est une allocation bien définie (car M est un modèle de $\Phi_\mathcal{P}$), et $\overrightarrow{\pi}'(i) \in \mathscr{G}_i$ pour tout $\varphi_i^* \in \mathscr{S}$. Puisque $\{\varphi_i^* \mid F(\overrightarrow{\pi}) \vDash \varphi_i^*\} \subset \mathscr{S}$, on a $Sat(\overrightarrow{\pi}') \supsetneq Sat(\overrightarrow{\pi})$. Donc $\overrightarrow{\pi}$ est Pareto-dominé. ▲

Ce résultat simple suggère que les partages efficaces peuvent être calculés à partir de l'ensemble de formules logiques $\Phi_\mathcal{P}$ du problème, plus précisément en calculant les sous-ensembles maximaux $\Gamma_\mathcal{P}$-consistants de $\Phi_\mathcal{P}$; nous les appellerons $\{\mathscr{S}_1, \dots, \mathscr{S}_q\}$. Pour chaque $\mathscr{S}_i$, soit $M_i = Mod(\bigwedge \mathscr{S}_i \wedge \Gamma_\mathcal{P})$ et soit $M = \cup_{i=1}^q M_i$. Alors $F^{-1}(M)$ est l'ensemble de toutes les allocations efficaces pour $\Phi_\mathcal{P}$. On peut remarquer qu'il y a en général un nombre exponentiel de sous-ensembles maximaux $\Gamma_\mathcal{P}$-consistants de $\Phi_\mathcal{P}$ (et donc en conséquence un nombre exponentiel de partages efficaces). Cette affirmation peut être néanmoins tempérée, car (a) il y a en pratique de nombreux cas pour lesquels le nombre de sous-ensembles maximaux consistants est faible ; (b) on ne cherche généralement pas *toutes* les allocations efficaces ; si l'on n'en cherche qu'une seule, elle peut être trouvée par calcul d'un sous-ensemble maximal $\Gamma_\mathcal{P}$-consistant de $\Phi_\mathcal{P}$.

Exemple 3.14.c Dans l'exemple 3.14, les sous-ensembles maximaux $\Gamma_\mathcal{P}$-consistants de $\Phi_\mathcal{P}$ sont $\mathscr{S}_1 = \{\varphi_1^*, \varphi_2^*\}$, $\mathscr{S}_2 = \{\varphi_1^*, \varphi_3^*\}$ et $\mathscr{S}_3 = \{\varphi_2^*, \varphi_3^*\}$. $\bigwedge \mathscr{S}_1 \wedge \Gamma_\mathcal{P}$ n'a qu'un seul modèle : $\{\mathbf{alloc(o_2, 1)}, \mathbf{alloc(o_3, 1)}, \mathbf{alloc(o_1, 2)}\}$. $\bigwedge \mathscr{S}_2 \wedge \Gamma_\mathcal{P}$ a deux modèles : $\{\mathbf{a_1}, \mathbf{alloc(o_2, 3)}\}$ et $\{\mathbf{alloc(o_2, 1)}, \mathbf{alloc(o_3, 1)}, \mathbf{alloc(o_1, 3)}\}$. $\bigwedge \mathscr{S}_3 \wedge \Gamma_\mathcal{P}$ a un seul modèle : $\{\mathbf{alloc(o_1, 2)}, \mathbf{alloc(o_2, 3)}\}$. En conséquence, les quatre allocations efficaces pour $\mathcal{P}$ sont $(\{o_2, o_3\}, \{o_1\}, -)$, $(\{o_1\}, -, \{o_2\})$, $(\{o_2, o_3\}, -, \{o_1\})$ et $(-, \{o_1\}, \{o_2\})$. Aucune d'entre elles n'est sans envie.

3.3.1.3 Partages efficaces et sans-envie

Rassemblons maintenant toutes les notions introduites. Puisque les partages sans envie correspondent aux modèles de $\Lambda_\mathcal{P}$ et que les allocations efficaces correspondent aux modèles des sous-ensembles maximaux $\Gamma_\mathcal{P}$-consistant de $\Phi_\mathcal{P}$, la propriété l'existence d'un partage efficace et sans-envie (EEF pour *Efficient and Envy-Free*) est équivalente à la condition suivante : *il existe un sous-ensemble $\mathscr{S}$ maximal $\Gamma_\mathcal{P}$-consistant de $\Phi_\mathcal{P}$ tel que $\bigwedge \mathscr{S} \wedge \Gamma_\mathcal{P} \wedge \Lambda_\mathcal{P}$ est consistant.* Dans ce cas, les modèles de cette dernière formule correspondent aux partages EEF. De manière intéressante, il s'agit d'une instance d'un problème connu dans le domaine du raisonnement non-monotone :

Définition 3.32 *Une* théorie des défauts supernormale[7] *est un couple $D = \langle\beta, \Delta\rangle$ avec $\Delta = \{\alpha_1, \dots, \alpha_m\}$, tel que $\alpha_1, \dots, \alpha_m$ et β sont des formules propositionnelles. Une formule proposi-*

[7] Les défauts «supernormaux» sont aussi appelés «défauts normaux sans prérequis» (voir par exemple [Reiter, 1980])

tionnelle ψ est une conséquence sceptique de D*, ce qui est noté* $D \mathrel{|\!\sim}^{\forall} \psi$*, si et seulement si pour tout* $\mathscr{S} \in MaxCons(\Delta, \beta)$ *on a* $\bigwedge \mathscr{S} \wedge \beta \models \psi$.

Une théorie des défauts est un cadre logique dédié à la modélisation de tâches de raisonnement du type «lois générales à exceptions». De manière informelle, l'ensemble des défauts supernormaux correspond à un ensemble de formules logiques, que nous tenons pour vraies si rien ne nous en empêche. Le β de la théorie est une formule logique qui est appelée un «fait», tenu pour vrai. Sa présence nous amène à nous interroger sur l'ensemble des formules de notre base de croyance Δ qui sont cohérentes avec le fait. Les sous-ensembles maximaux β-consistants de Δ sont les sous-ensembles maximaux dans notre base de croyance que le fait ne «remet pas en cause». La question de l'inférence sceptique correspond donc à déterminer si l'on peut déduire une formule à partir du fait β et en changeant le moins possible l'ensemble de nos croyances.

Le lien avec l'absence d'envie et l'efficacité dans les problèmes de partage n'est pas évident. Et pourtant, nous avons la proposition suivante :

Proposition 3.6 *Soit* $\mathcal{P} = (\mathscr{N}, \mathscr{O}, (\varphi_1, \ldots, \varphi_n))$ *une instance du problème de partage équitable avec préférences dichotomiques. Soit* $D_P = \langle \Gamma_{\mathcal{P}}, \Phi_{\mathcal{P}} \rangle$*. Alors il existe un partage efficace et sans envie pour* $\mathcal{P}$ *si et seulement si* $D \not\mathrel{|\!\sim}^{\forall} \neg\Lambda_{\mathcal{P}}$.

Démonstration Soit $\mathcal{P} = (\mathscr{N}, \mathscr{O}, (\varphi_1, \ldots, \varphi_n))$ une instance du problème de partage équitable avec préférences dichotomiques. Soit $\overrightarrow{\pi}$ une allocation Pareto-efficace et sans envie, et $\mathscr{S} = \{\varphi_i^* \mid F(\overrightarrow{\pi}) \models \varphi_i^*\}$. Alors $F(\overrightarrow{\pi}) \models \Lambda_{\mathcal{P}}$ d'après la proposition 3.4. Nous avons aussi $F(\overrightarrow{\pi}) \models \mathscr{S}$ par définition de $\mathscr{S}$, et donc $F(\overrightarrow{\pi}) \models \bigwedge \mathscr{S} \wedge \Lambda_{\mathcal{P}}$, ce qui prouve que $\bigwedge \mathscr{S} \wedge \Lambda_{\mathcal{P}}$, ou en d'autres termes, que $\bigwedge \mathscr{S} \not\models \neg\Lambda_{\mathcal{P}}$. De plus, $\mathscr{S}$ est un sous-ensemble maximal $\Gamma_{\mathcal{P}}$-consistant de $\Phi_{\mathcal{P}}$ d'après la proposition 3.5. Ainsi $\mathscr{S} \in MaxCons(\Phi_{\mathcal{P}}, \Gamma_{\mathcal{P}})$, et $\bigwedge \mathscr{S} \wedge \Gamma_{\mathcal{P}} \not\models \neg\Lambda_{\mathcal{P}}$, ce qui implique $\langle \Gamma_{\mathcal{P}}, \Phi_{\mathcal{P}} \rangle \not\mathrel{|\!\sim}^{\forall} \neg\Lambda_{\mathcal{P}}$ par définition de $\not\mathrel{|\!\sim}^{\forall}$.

Réciproquement, supposons que $\langle \Gamma_{\mathcal{P}}, \Phi_{\mathcal{P}} \rangle \not\mathrel{|\!\sim}^{\forall} \neg\Lambda_{\mathcal{P}}$. Alors il existe un ensemble $\mathscr{S} \in MaxCons(\Phi_{\mathcal{P}}, \Gamma_{\mathcal{P}})$ tel que $\bigwedge \mathscr{S} \wedge \Gamma_{\mathcal{P}} \wedge \Lambda_{\mathcal{P}}$ a un modèle M. D'après la proposition 3.5, $F^{-1}(M)$ est un partage Pareto-efficace (puisque M est un modèle de $\bigwedge \mathscr{S} \wedge \Gamma_{\mathcal{P}}$), et d'après la proposition 3.4, $F^{-1}(M)$ est sans envie (puisque M est un modèle de $\Lambda_{\mathcal{P}}$). ▲

Ce lien plutôt inattendu avec le raisonnement non monotone a de nombreuses implications. Tout d'abord, les allocations EEF correspondent aux modèles de $\bigwedge \mathscr{S} \wedge \Gamma_{\mathcal{P}} \wedge \Lambda_{\mathcal{P}}$ avec $\mathscr{S} \in MaxCons(\Phi_{\mathcal{P}}, \Gamma_{\mathcal{P}})$. Cependant, l'ensemble $MaxCons(\Phi_{\mathcal{P}}, \Gamma_{\mathcal{P}})$ peut être exponentiellement large, ce qui nous dissuade de commencer par calculer les allocations efficaces et de filtrer ensuite celles qui ne sont pas sans envie, et plaide plutôt en faveur du calcul des allocations EEF en une seule passe, en utilisant des algorithmes issus du raisonnement à base de défauts. Ainsi, le partage équitable peut éventuellement bénéficier du travail algorithmique sur la logique des défauts et sur des domaines connexes tels que la révision des croyances. De plus, les critères alternatifs pour la sélection des extensions en raisonnement à base de défauts (tels que la cardinalité, les poids ou les priorités) correspondent à une alternative à la propriété d'efficacité dans les problèmes de partage.

3.3.1.4 Conclusion sur le problème avec préférences dichotomiques

Nous avons donc introduit dans cette sous-section un langage de représentation compacte dédié à la représentation compacte d'un problème de partage de biens indivisibles avec des préférences dichotomiques. Ce langage est fondé sur la définition d'une instance comme un ensemble d'agents, un ensemble d'objets et un ensemble de formules logiques. Dans ce cadre, les propriétés d'absence d'envie et de Pareto-efficacité s'expriment de manière logique, ce qui nous a permis de dresser un parallèle intéressant entre le problème d'existence d'un partage Pareto-efficace et sans envie et le problème d'inférence sceptique en raisonnement des défauts. Ce parallèle sera à la base de l'analyse de la complexité de ce problème, que nous effectuerons au chapitre 4.

3.3.2 Un langage générique pour le problème de partage de biens indivisibles

Le précédent langage que nous avons introduit était dédié à la représentation compacte de propriétés ordinales du partage telles que la Pareto-efficacité et l'absence d'envie. Les préférences dichotomiques se prêtaient bien à cette modélisation : d'une part ce sont les structures de préférence ordinales les plus basiques, d'autres part, elles se prêtent bien à la représentation compacte par le biais de la logique propositionnelle, et enfin, comme nous le verrons au chapitre suivant, elles permettent d'aborder de manière simple le problème de la complexité liée à la représentation compacte, à l'absence d'envie et à la Pareto-efficacité.

Nous allons maintenant nous intéresser à la définition d'un langage générique pour représenter les problèmes de partage de biens indivisibles. L'objectif de ce langage est d'être intuitif et expressif. Nous allons nous fonder pour sa définition sur l'ensemble des notions introduites au chapitre 1, notamment sur la définition générique d'une instance du problème de partage de biens indivisibles (voir la définition 1.37).

Le langage de représentation compacte que nous allons introduire ici est fondé sur :
- ▷ une représentation logique des contraintes d'admissibilité ;
- ▷ des agents ayant des préférences numériques exprimées sous la forme de formules pondérées ;
- ▷ une modélisation *welfariste* du bien-être social, c'est-à-dire à l'aide d'une fonction d'utilité collective.

Ce langage de représentation a été décrit en détails dans [Fargier *et al.*, 2004a; Bouveret *et al.*, 2005a,b]. Il est fondé sur les mêmes éléments de base que dans la définition 1.37, c'est-à-dire :
- ▷ un ensemble fini d'agents $\mathscr{N} = \{1, \ldots, n\}$;
- ▷ un ensemble fini d'objets $\mathscr{O}$, de taille p .

De même, on définit toujours un partage comme un vecteur $\overrightarrow{\pi} = (\pi_1, \ldots, \pi_n)$, où $\pi_i \subseteq \mathscr{O}$ est la part de l'agent i.

3.3.2.1 Contraintes

Dans le chapitre 1, nous avons supposé que la spécification des alternatives admissibles se faisait à l'aide d'un ensemble de contraintes, définies — dans la définition 1.3 — comme un ensemble de tuples de $\mathscr{O}^n$. Nous avons vu dans ce chapitre qu'il existait un moyen simple de représenter un tel ensemble, par le biais de la logique propositionnelle et du langage $\mathcal{L}_{\mathscr{O}}^{alloc}$. Nous spécifierons donc dans notre langage l'ensemble des partages admissibles à l'aide d'un ensemble de contraintes exprimées sous forme logique :

Définition 3.33 (Contrainte, partage admissible) *Une contrainte est une formule de $L_{\mathscr{O}}^{alloc}$.*

L'ensemble des partages admissibles *vis-à-vis d'un ensemble de contraintes $\mathscr{C}$ exprimées comme des formules de $L_{\mathscr{O}}^{alloc}$ est l'ensemble $\mathscr{A} = \{\overrightarrow{\pi} \mid \forall C \in \mathscr{C},\ h^{-1}(\overrightarrow{\pi}) \models C\}$, avec h la bijection introduite dans la preuve de la proposition 3.1 page 84.*

Ce langage permet de représenter de manière compacte la plupart des contraintes usuelles :
- ▷ La contrainte de préemption globale s'exprime comme un ensemble de $n \times (n-1) \times p$ contraintes (ou comme une contrainte unique, comme nous l'avons vu dans le langage dédié aux préférences dichotomiques) : $\forall (i, j, o) \in \mathscr{N}^2 \times \mathscr{O}$ tels que $i \neq j$, $\neg(\mathbf{alloc(o, i)} \wedge \mathbf{alloc(o, j)})$.

- ▷ Une contrainte d'exclusion sur un ensemble d'objets $\mathscr{S}_{excl}$ s'exprime comme un ensemble de n contraintes : $\forall i \in \mathscr{N}$, $\neg \bigwedge_{o \in \mathscr{S}_{excl}} \mathbf{alloc(o, i)}$.
- ▷ Plus généralement, on peut exprimer facilement des contraintes de dépendance entre les objets à l'aide de formules logiques générales. Par exemple, $\neg\mathbf{alloc(o_1, i)} \vee (\mathbf{alloc(o_1, i)} \wedge (\mathbf{alloc(o_2, i)} \vee \mathbf{alloc(o_3, i)}))$ exprime le fait que si un objet o_1 est attribué à l'agent i, alors cet agent doit recevoir l'un des objets o_2 et o_3.

S'il est facile d'exprimer la plupart des contraintes usuelles dans ce langage de représentation, il est en revanche difficile (et, de manière générale, impossible) de représenter des contraintes de volume de manière compacte dans ce langage. De telles contraintes pourraient bien entendu être représentées par un ensemble de contraintes d'exclusion, mais une telle traduction peut nécessiter un nombre de contraintes exponentiel en p. Lors de notre analyse des classes de complexité dans le chapitre 4, nous nous autoriserons ponctuellement l'extension de ce langage de représentation de contraintes permettant l'expression de contraintes de volume. Nous introduirons la même extension dans l'implantation informatique du modèle, présentée dans le chapitre 6.

Ce langage d'expression de contraintes amène une autre remarque, liée à la nature des contraintes elle-même. Nous avons défini la notion de contrainte comme un sous-ensemble de l'ensemble des partages admissibles, autorisant ainsi la définition de contraintes s'appliquant spécifiquement à des agents. Il est naturel de s'interroger sur la pertinence de cette définition, alors que la plupart des contraintes naturelles s'appliquent de manière homogène à tous les agents (par exemple, des contraintes physiques ou légales interdisant l'attribution simultanée de deux objets au même agent s'appliquent probablement à tous les agents), et donc pourraient être exprimées de manière plus naturelle et plus compacte sur le langage $\mathcal{L}_{\mathscr{O}}$, et non sur l'ensemble des partages possibles, langage pour lequel l'expression de telles contraintes nécessite l'introduction de n contraintes (ou d'une contrainte impliquant une conjonction de taille n) identiques au numéro agent près (voir par exemple l'expression d'une contrainte d'exclusion ci-dessus). Nous avons cependant fait le choix d'une plus grande expressivité en autorisant l'expression de contraintes s'appliquant spécifiquement à certains agents, sachant que ce choix, s'il implique une représentation légèrement moins compacte que si nous avions exprimé des contraintes sur l'ensemble des lots, n'entraîne pas une augmentation exponentielle de la taille de la représentation, et plus généralement, n'entraîne pas d'augmentation de la complexité du problème de partage.

3.3.2.2 Demandes pondérées et préférences individuelles

L'espace des alternatives étant défini, il nous faut maintenant spécifier la manière dont les agents expriment leurs préférences. Dans la modélisation que nous avons choisie, qui s'inspire du *welfarisme*, les préférences des agents sont représentées par des fonctions d'utilité. Il reste à préciser comment sont construites ces fonctions d'utilité.

Parmi les langages de représentation de préférences présentés ci-avant, le langage $\mathcal{R}_{weighted}$ à base de buts pondérés (définition 3.18) fournit un bon compromis entre la concision, la puissance d'expression, et l'utilisation intuitive. Nous considérerons donc que les préférences des agents sont exprimées sous la forme d'une base de buts pondérée sur le langage propositionnel $\mathcal{L}_{\mathscr{O}}$ fondé sur l'ensemble d'objets. Nous limiterons cependant le langage à l'agrégation des demandes satisfaites, et nous ne prendrons donc pas en compte les demandes non satisfaites.

Définition 3.34 (Demande pondérée) *Une* demande pondérée *est un couple $\delta = (\varphi(\delta), w(\delta))$, avec :*

- ▷ *$\varphi(\delta)$ une formule de $L_{\mathscr{O}}$,*
- ▷ *$w(\delta) \in \mathscr{V}_{ind}$, où $\langle \mathscr{V}_{ind}, \succeq_{ind}, \oplus \rangle$ est un espace de valuation ordonné par $\succeq_{ind}$, et dont la loi $\oplus$ respecte toutes les conditions de la définition 1.12.*

L'ensemble des demandes pondérées de l'agent i est noté Δ_i.

Par la suite, nous ferons l'hypothèse de *monotonie* des préférences individuelles (voir définition 1.13 à la page 26), vis-à-vis de la loi $\oplus$ et de l'ordre $\succeq_{ind}$, c'est-à-dire telles que $\forall(\pi, \pi')$, $\pi \subseteq \pi' \Rightarrow u(\pi) \preceq u(\pi')$. Concrètement cela revient à restreindre les ensembles Δ_i aux demandes δ telles que $\varphi(\delta) \in \mathcal{L}^+_{\mathscr{O}}$ et $w(\delta)$ est un poids positif, c'est-à-dire tel que $\forall a \in \mathscr{V}_{ind}$, $a \oplus w(\delta) \succeq_{ind} a$.

Tout comme dans le langage de la définition 3.18, le calcul de l'utilité d'un agent est effectué par agrégation des poids des demandes satisfaites. Les définitions suivantes décrivent de manière plus précise le processus de calcul de l'utilité individuelle d'un agent à partir de la part qu'il reçoit.

Définition 3.35 (Valeur de satisfaction d'une demande) *Soient π un ensemble d'objets et δ une demande pondérée. On dira que δ est* satisfaite *par π si et seulement si $\pi \vDash \varphi(\delta)$. La* valeur de satisfaction *de δ vis-à-vis de π est définie comme suit :*

$$\sigma(\delta, \pi) = \begin{cases} w(\delta) & \text{si } \delta \text{ est satisfaite par } \pi, \\ \bot & \text{sinon.} \end{cases}$$

Définition 3.36 (Utilité individuelle) *Étant donnés un agent i, son ensemble de demandes pondérées Δ_i, et sa part π_i, l'*utilité individuelle *de l'agent i est définie comme suit :*

$$u_i(\pi_i) = \bigoplus_{\delta \in \Delta_i} \sigma(\delta, \pi_i).$$

La loi $\oplus$ sera appelé fonction d'agrégation individuelle.

Commentons cette définition. Premièrement, elle implique que la satisfaction d'un agent ne dépend que de ce qu'il reçoit. Nous pouvons donc écrire $u_i(\overrightarrow{\pi}) = u_i(\pi_i)$. La croissance des demandes vis-à-vis de la loi $\oplus$ signifie que le fait de satisfaire une demande ne peut jamais avoir un impact négatif (sur la satisfaction d'un agent), la commutativité de la loi $\oplus$ implique que l'ordre dans lequel sont faites des demandes n'est pas significatif, et l'associativité de cette même loi (due au fait qu'il s'agit d'une loi interne binaire) permet de calculer l'utilité d'un agent de manière simple, et ce même quand l'ensemble des demandes est énorme.

Le choix le plus évident pour la fonction d'agrégation $\oplus$ qui apparaît dans la définition 3.36 est $\oplus = +$; cependant, il peut y avoir des raisons, dans certains cas, de choisir $\oplus = \max$. Premièrement, $\oplus = \max$ est appropriée dans les cas où les agents ne veulent pas plus d'un objet. Bien entendu, de telles fonctions d'utilité pourraient être exprimées en utilisant $\oplus = +$, mais leur expression serait bien moins compacte. De plus, certains problèmes ont une complexité moindre avec $\oplus = \max$ qu'avec $\oplus = +$, comme nous allons le voir au chapitre 4. Deuxièmement, dans certains cas, les agents ne veulent pas (ou ne peuvent pas) exprimer leurs préférences numériquement, et en conséquence sont plus enclins à exprimer des listes de priorité. Dans ce dernier cas cependant, il serait peut-être plus judicieux d'utiliser un espace $\mathscr{V}_{ind}$ vectoriel, et d'utiliser un préordre leximax pour comparer les vecteurs des poids des formules satisfaites.

3.3.2.3 Utilité collective

Tout comme dans le modèle introduit au chapitre 1 fondé sur la théorie de l'utilitarisme, la comparaison des partages se fait sur la base des profils d'utilité et de l'utilité collective déduite de ces profils d'utilité.

Définition 3.37 (Utilité collective) *Soient $\mathscr{N}$ un ensemble d'agents et $\overrightarrow{\pi}$ un partage. L'utilité collective des agents vis-à-vis du partage $\overrightarrow{\pi}$ est définie comme suit :*

$$u_c(\overrightarrow{\pi}) = g(u_1(\overrightarrow{\pi}), \dots, u_n(\overrightarrow{\pi})),$$

avec g une fonction de $\mathscr{V}_{ind}^n$ dans $\mathscr{V}$, espace de valuation ordonné par $\succeq$, non décroissante par rapport à chaque composante, c'est-à-dire telle que :

$$\forall (u_1, \dots, u_{i-1}, u_i, u_{i+1}, \dots, u_n) \in \mathscr{V}_{ind}^n \text{ et } \forall u'_i \succeq_{ind} u_i,$$
$$g(u_1, \dots, u_{i-1}, u'_i, u_{i+1}, \dots, u_n) \succeq g(u_1, \dots, u_{i-1}, u_i, u_{i+1}, \dots, u_n).$$

Notons que cette définition générique autorise toutes les fonctions d'utilité à valeurs réelles introduites dans le chapitre 1, mais permet aussi de représenter de manière explicite des ordres de bien-être social tels que l'ordre leximin, en choisissant $\mathscr{V} = \mathscr{V}_{ind}^n$, $\succeq = \succeq_{leximin}$, et $g = Id$. Notons de plus que la généricité de la fonction g autorise en particulier toutes les fonctions d'utilité collective à droits exogènes inégaux, introduites au chapitre 2.

L'utilité collective associée à un partage admissible $\overrightarrow{\pi}$ est donc calculée par deux agrégations successives : agrégation des poids des demandes individuelles en utilités individuelles, puis agrégation des utilités individuelles en utilités collectives :

$$\left.\begin{array}{ccccccc}
\sigma(\delta_{1,1}, \pi_1) & \oplus & \dots & \oplus & \sigma(\delta_{1,m_1}, \pi_1) & \rightsquigarrow & u_1(\pi_1) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\sigma(\delta_{n,1}, \pi_n) & \oplus & \dots & \oplus & \sigma(\delta_{n,m_n}, \pi_n) & \rightsquigarrow & u_n(\pi_n)
\end{array}\right\} \rightsquigarrow u_c(\overrightarrow{\pi}) = g(u_1(\pi_1), \dots, u_n(\pi_n)),$$

avec bien entendu m_i désignant le nombre de demandes de l'agent i.

3.3.2.4 Problème de partage de biens indivisibles générique

Nous avons maintenant tous les éléments nécessaires à la définition d'une instance du problème de partage de biens indivisibles générique dans notre langage de représentation compacte :

Définition 3.38 (Instance du problème de partage de biens indivisibles générique) *Une instance du problème de partage de biens indivisibles générique est un tuple $(\mathscr{N}, \mathscr{O}, (\Delta_1, \dots, \Delta_n), \mathscr{C}, \langle \mathscr{V}_{ind}, \succeq_{ind}, \oplus \rangle, \langle \mathscr{V}, \succeq \rangle, g)$, où :*

- ▷ *$\mathscr{N} = \{1, \dots, n\}$ est un ensemble fini d'agents ;*
- ▷ *$\mathscr{O}$ un ensemble fini d'objets ;*
- ▷ *$\Delta_i \subset L_{\mathscr{O}}^{+} \times \mathscr{V}_{ind}$ est un ensemble fini de demandes pondérées pour chaque agent i ;*
- ▷ *$\mathscr{C} \subset L_{\mathscr{O}}^{alloc}$ est un ensemble de contraintes ;*
- ▷ *$\langle \mathscr{V}_{ind}, \succeq_{ind}, \oplus \rangle$ est la structure de valuation individuelle, vérifiant les conditions de la définition 1.12, avec $\oplus$ la fonction d'agrégation individuelle ;*
- ▷ *$\langle \mathscr{V}, \succeq \rangle$ est la structure de valuation collective, avec g la fonction d'utilité collective.*

Définition 3.39 (Solution du problème de partage) *Une solution du problème de partage est un partage admissible $\overrightarrow{\pi}$.*

L'utilité collective associée à une solution $\overrightarrow{\pi}$ du problème de partage est calculée par double agrégation des poids des demandes et des utilités individuelles :

$$u_c(\overrightarrow{\pi}) = g\left(\bigoplus_{\delta \in \Delta_1} \sigma(\delta, \pi_1), \dots, \bigoplus_{\delta \in \Delta_n} \sigma(\delta, \pi_n)\right).$$

3.3.2.5 Traduction des problèmes d'enchères combinatoires

Il est naturel de se demander dans quelle mesure il est possible de représenter de manière concise dans notre cadre générique les problèmes de partage classiques que sont les problèmes d'enchères combinatoires, représentés de manière compacte à l'aide des langages de lots (OR, XOR et autres langages dérivés). En fait, il est possible d'utiliser notre cadre générique sans perte de concision pour représenter des problèmes typiques des enchères combinatoires, ce qui rend ce modèle encore plus intéressant :

Proposition 3.7 *Toute instance du Winner Determination Problem dans les enchères combinatoires, pour laquelle les préférences sont exprimées à l'aide du langage OR ou XOR peut être traduite en une instance équivalente de notre problème de partage de biens indivisibles de taille polynomiale en la taille de l'instance initiale.*

Par «équivalente», nous entendons le fait que toute solution optimale initiale du *Winner Determination Problem* est une solution optimale (maximisant l'utilité collective) de notre problème de partage, et vice-versa.

Démonstration Soit $\mathcal{P}_{OR} = (\mathscr{N}_{OR}, \mathscr{O}_{OR}, (\Delta_{1,OR}, \ldots, \Delta_{n,OR}))$ une instance du *Winner Determination Problem* dont les mises sont exprimées dans le langage OR. Pour chaque mise atomique (π_i^j, w_i^j) dans $\Delta_{i,OR}$, nous introduisons un objet supplémentaire α_i^j qui nous servira à empêcher qu'un agent ne reçoive à la fois l'utilité de deux mises dont les lots correspondants ont une intersection non vide.

L'instance $\mathcal{P}_{OR}$ est traduite en une instance $\mathcal{P}_{PBI} = (\mathscr{N}, \mathscr{O}, (\Delta_1, \ldots, \Delta_n), \mathscr{C}, \langle \mathscr{V}_{ind}, \succeq_{ind}, \oplus \rangle, \langle \mathscr{V}, \succeq, \rangle, g)$ du problème de partage de biens indivisibles, où :

- ▷ $\mathscr{N} = \mathscr{N}_{OR}$;
- ▷ $\mathscr{O} = \mathscr{O}_{OR} \cup \bigcup_{i \in \mathscr{N},\ (\pi_i^j, w_i^j) \in \Delta_{i,OR}} \alpha_i^j$;
- ▷ $\forall i,\ \Delta_i = \{\langle \alpha_i^j \wedge \bigwedge_{o_k \in \pi_i^j} \mathbf{o_k}, w_i^j \rangle \mid (\pi_i^j, w_i^j) \in \Delta_{i,OR}\}$
- ▷ $\mathscr{C} = \{\neg \mathbf{alloc}(\alpha_\mathbf{i}^{\mathbf{j_1}}, \mathbf{i}) \wedge \mathbf{alloc}(\alpha_\mathbf{i}^{\mathbf{j_2}}, \mathbf{i}) \mid i \in \mathscr{N},\ (\pi_i^{j_1}, w_i^{j_1}) \in \Delta_i,\ (\pi_i^{j_2}, w_i^{j_2}) \in \Delta_i \text{ et } \pi_i^{j_1} \cap \pi_i^{j_2} \neq \varnothing\} \cup C_{preempt}$;
- ▷ $\langle \mathscr{V}_{ind}, \succeq_{ind}, \oplus \rangle = \langle \mathbb{R}^+, \succeq, + \rangle$;
- ▷ $\langle \mathscr{V}, \succeq, \rangle = \langle \mathbb{R}^+, \succeq \rangle$;
- ▷ $g : (u_1, \ldots, u_n) \mapsto \sum_{i=1}^n u_i$.

Il est clair que $\mathcal{P}_{PBI}$ peut être obtenue à partir de $\mathcal{P}_{OR}$ en temps polynomial. Nous avons maintenant à prouver que toutes les solutions optimales de $\mathcal{P}_{OR}$ correspondent à des solutions optimales de $\mathcal{P}_{PBI}$ et vice-versa.

Soit $\overrightarrow{\pi}_{OR}$ une solution optimale du *Winner Determination Problem* associé à $\mathcal{P}_{OR}$, et pour tout $i \in \mathscr{N}$, soit $\mathscr{W}_i$ une partition de $\pi_{i,OR}$ qui satisfait $\mathscr{W}_i = \underset{\mathscr{W} \text{ partition de } \pi_{i,OR}}{\operatorname{argmax}} \sum_{\pi_i^j \in \mathscr{W}} w_i^j$ (en d'autres termes, la partition $\mathscr{W}_i$ sélectionne l'ensemble des mises de l'agent i dont les valeurs vont contribuer à l'utilité de cet agent). Alors l'allocation $\overrightarrow{\pi}$ de $\mathcal{P}_{PBI}$ telle que $\forall i$, $\pi_i = \pi_{i,OR} \cup \bigcup_{\pi_i^j \in \mathscr{W}_i} \alpha_i^j$ est admissible. En effet, puisque les $\mathscr{W}_i$ sont des partitions, $\overrightarrow{\pi}$ satisfait les contraintes d'exclusion de $\mathscr{C}$, et bien sûr aussi la contrainte de préemption. De plus, on peut vérifier aisément que pour tout agent i, $u_i(\overrightarrow{\pi}) = u_i(\overrightarrow{\pi}_{OR})$, et donc $u_c(\overrightarrow{\pi}) = u_c(\overrightarrow{\pi}_{OR})$ par définition des opérateurs d'agrégation de $\mathcal{P}_{PBI}$.

Maintenant supposons qu'il existe une solution $\overrightarrow{\pi}'$ de $\mathcal{P}_{PBI}$ telle que $u_c(\overrightarrow{\pi}') > u_c(\overrightarrow{\pi})$. Soit alors $\overrightarrow{\pi}'_{OR}$ l'allocation de $\mathcal{P}_{OR}$ telle que toute part $\pi'_{i,OR}$ d'un agent i correspond à la part π'_i restreinte à $\mathscr{O}_{OR}$. Pour chaque agent i, l'ensemble $\mathscr{W}'_i = \bigcup_{j \text{ tel que } \alpha_i^j \in \pi'_i} \pi_i^j$ est une partition de π'_i. Donc l'utilité individuelle de i dans $\mathcal{P}_{OR}$ est supérieure ou égale à $\sum_{\pi_i^j \in \mathscr{W}'_i} w_i^j = u'_i$, l'utilité individuelle de l'agent i dans $\mathcal{P}_{PBI}$. Nous avons donc $u_c(\overrightarrow{\pi}'_{OR}) \succeq u_c(\overrightarrow{\pi}')$, donc $u_c(\overrightarrow{\pi}'_{OR}) > u_c(\overrightarrow{\pi})$ (avec l'hypothèse de départ que $u_c(\overrightarrow{\pi}') > u_c(\overrightarrow{\pi})$), et donc finalement $u_c(\overrightarrow{\pi}'_{OR}) > u_c(\overrightarrow{\pi}_{OR})$, ce qui contredit le fait que $\overrightarrow{\pi}_{OR}$ est une solution optimale de $\mathcal{P}_{OR}$. Cela prouve au final que $\overrightarrow{\pi}$ est une solution optimale de $\mathcal{P}_{PBI}$.

À l'inverse, la construction d'une solution de $\mathcal{P}_{OR}$ à partir d'une solution $\overrightarrow{\pi}$ de $\mathcal{P}_{PBI}$ peut se faire en projetant $\overrightarrow{\pi}$ la première sur l'ensemble $\mathscr{O}_{OR}$. La preuve est relativement similaire à ce qui précède, donc nous l'omettrons.

En ce qui concerne le langage XOR, on peut vérifier aisément que le remplacement de l'opérateur d'agrégation individuel $\oplus = +$ par $\oplus = \max$ dans la réduction précédente permet d'appliquer cette même réduction à une instance du *Winner Determination Problem* pour laquelle les préférences des agents sont exprimées dans le langage XOR. Pour cette réduction, l'introduction des objets α_i^j n'est même pas nécessaire, car l'opérateur max se charge de rendre les lots mutuellement exclusifs pour le calcul de l'utilité individuelle des agents. ▲

La proposition 3.7 a pour conséquence directe le fait qu'il est possible de traduire en temps et espace polynomiaux toute instance du problème d'enchère combinatoire dont les préférences des agents sont exprimées dans tout langage dérivé des langages OR et XOR (OR-of-XOR, OR*, OR / XOR, etc.) en une instance de notre modèle. En effet, il suffit de traduire dans un premier temps les mises des agents dans le langage OR* ([Nisan, 2000] nous confirme que cette traduction peut se faire en n'utilisant qu'un nombre polynomial d'objets factices), puis de traiter ces mises de la même manière que pour la traduction détaillée dans la preuve de la proposition 3.7.

3.4 Conclusion

Dans ce chapitre, nous avons mis en valeur le phénomène d'explosion combinatoire et défini quelques notions ayant trait à la représentation formelle compacte d'un espace d'alternatives combinatoire (autrement dit, des contraintes restreignant l'ensemble des alternatives admissibles), et des préférences sur cet espace. Nous avons ensuite présenté une vue d'ensemble des langages de représentation compacte classiques utilisés dans la littérature, d'une part pour la représentation des contraintes d'admissibilité de l'espace des alternatives, et d'autre part pour la représentation des préférences des agents sur cet espace.

Nous avons enfin appliqué ces notions à la représentation compacte de deux problèmes de partage de biens indivisibles. Dans un premier temps, nous nous sommes intéressés à la représentation compacte du problème de partage de biens indivisibles avec des préférences dichotomiques, et nous avons montré que les propriétés d'absence d'envie et de Pareto-efficacité pouvaient s'exprimer sous forme logique. Dans un deuxième temps, nous avons défini un cadre formel de représentation compacte dédié au problème de partage de biens indivisibles tel que nous l'avons introduit au chapitre 1 dans la définition 1.37. Ce cadre formel est fondé sur une représentation logique des contraintes d'admissibilité et des préférences des agents. Il s'inspire de la théorie du *welfarisme* pour l'agrégation de ces préférences.

La suite logique de ce travail sur la représentation compacte des problèmes de partage de biens indivisibles est l'analyse de leur complexité théorique. Nous nous pencherons sur cette question dans le chapitre 4. Le chapitre 5 sera quant à lui consacré aux aspects algorithmiques liés au calcul de solutions optimales au sens du bien-être social, dans le cas particulier de l'ordre de bien-être collectif leximin.

Chapitre 4
Complexité du problème de partage

Comme nous l'avons vu au chapitre précédent, l'introduction de la représentation compacte dans les problèmes de partage est un moyen remarquable de concilier la concision et l'expressivité dans la spécification des contraintes et des préférences. Cependant, l'emploi d'un langage d'expression compacte pose une question cruciale : celle de la complexité du problème de partage spécifié de cette manière. Jusqu'ici, les aspects liés à la complexité théorique des problèmes de partage n'ont été majoritairement étudiés que dans deux contextes précis : celui des enchères combinatoires, et celui de la négociation [Endriss et Maudet, 2004; Chevaleyre *et al.*, 2004; Dunne *et al.*, 2005]. En revanche, la complexité du problème de partage équitable de biens indivisibles n'a jusqu'à ce jour jamais été étudiée à notre connaissance, sauf dans [Lipton *et al.*, 2004], qui traite de l'existence de schémas d'approximation pour le problème de minimisation de l'envie dans une instance du problème de partage avec préférences additives.

Dans ce chapitre nous allons nous intéresser en détail à la complexité des deux problèmes introduits dans le chapitre précédent. Le premier problème est lié à la recherche de partages efficaces et sans envie en présence de préférences dichotomiques exprimées sous forme logique et sans autre contrainte que la contrainte de préemption. Le second problème concerne la maximisation de l'utilité collective, définie comme l'agrégation des utilités individuelles elles-mêmes définies par des formules logiques pondérées, le tout en présence de contraintes exprimées sous forme logique. Ces résultats ont été publiés respectivement dans [Bouveret *et al.*, 2005a,b] et [Bouveret et Lang, 2005].

Nous supposerons dans ce chapitre que les notions de base de la théorie de la complexité sont connues du lecteur, ainsi que quelques-unes des classes de problèmes les plus courantes. L'annexe A rappelle quelques notions et définitions de base nécessaires à la compréhension du chapitre ; on pourra de même consulter avec profit les ouvrages de référence tels que [Papadimitriou, 1994; Garey et Johnson, 1979].

4.1 Existence d'une allocation efficace et sans-envie

4.1.1 Complexité du problème EEF avec préférences dichotomiques

4.1.1.1 Le problème général

Intéressons nous à la complexité du problème d'existence d'une allocation Pareto-efficace et sans envie, tel qu'il a été défini dans la section 3.3.1.3, et que nous noterons [EEF EXISTENCE]. Nous avons vu dans la proposition 3.6 de la page 120 que l'existence d'un partage Pareto-efficace et sans envie dans ce langage pouvait se ramener au complémentaire d'un problème d'inférence sceptique en logique des défauts. La complexité de ce problème est bien connue [Gottlob, 1992] : il est Π_2^{p}-complet. Cela a pour implication immédiate que le problème d'existence d'un partage Pareto-efficace et sans envie est dans Σ_2^{p}. Nous allons maintenant démontrer que ce problème est complet pour cette classe, ce qui est moins évident, et ce, même si les préférences sont monotones. Afin de prouver la difficulté du problème, nous utiliserons la restriction suivante du problème d'inférence sceptique :

Problème 2: [RSI] (Restricted Skeptical Inference)

INSTANCE :	Un ensemble de formules propositionnelles $\Delta = \{\alpha_1, \dots, \alpha_n\}$.
QUESTION :	Tous les ensembles maximaux-consitants de Δ contiennent-ils α_1 ?

Proposition 4.1 *Le problème [RSI] est Π_2^{p}-complet.*

Démonstration L'appartenance à Π_2^{p} vient facilement du fait que le problème [RSI] est une restriction du problème d'inférence sceptique général, pour laquelle la formule φ à inférer est simplement α_1. La difficulté vient du fait que pour chaque instance (Δ, β, ψ) du problème d'inférence sceptique on a $(\Delta, \beta) \mid\!\sim^{\forall} \psi$ si et seulement si $(\{\psi\} \cup \Delta, \beta) \mid\!\sim^{\forall} \psi$, si et seulement si $(\{\psi \wedge \beta\} \cup \{\alpha_1 \wedge \beta, \dots, \alpha_n \wedge \beta\}, \top) \mid\!\sim^{\forall} \psi$, si et seulement si tous les sous-ensembles maximaux β-consistants de $\{\psi, \alpha_1, \dots, \alpha_n\}$ contiennent ψ, ce qui est une instance de [RSI]. ▲

Proposition 4.2 *Le problème [EEF EXISTENCE] qui consiste à déterminer s'il existe un partage efficace et sans envie pour une instance donnée $\mathcal{P}$ du problème de partage avec des préférences monotones dichotomiques sous forme logique est Σ_2^{p}-complet.*

La difficulté sera prouvée à l'aide de la réduction suivante depuis le problème $\overline{\text{[RSI]}}$ (le problème complémentaire de [RSI], c'est-à-dire, existe-t-il un sous-ensemble maximal consistant de Δ qui ne contient pas α_1 ?) vers le problème [EEF EXISTENCE]. Étant donné un ensemble fini Δ de formules propositionnelles, on notera par la suite $V_\Delta = Var(\Delta)$ l'ensemble des symboles propositionnels apparaissant dans Δ. Soit $\mathcal{P}(\Delta) = (\mathscr{N}, \mathscr{O}, \Phi_{\mathcal{P}(\Delta)})$ l'instance suivante de [EEF EXISTENCE] :

1. $\mathscr{N} = \{1, 2, ..., n+3\}$;
2. $\mathscr{O} = \{\mathbf{v^i} \mid \mathbf{v} \in V_\Delta, i \in 1...n\} \cup \{\mathbf{\bar{v}^i} \mid \mathbf{v} \in V_\Delta, i \in 1...n\} \cup \{\mathbf{x^i} \mid i \in 1...n+1\} \cup \{\mathbf{y}\}$;
3. pour tout $i = 1, \dots, n$, soit β_i la formule obtenue à partir de α_i par la séquence d'opérations suivante : (i) mettre α_i sous forme NNF (soit α'_i le résultat) ; (ii) pour tout $\mathbf{v} \in V_\Delta$, remplacer, dans α'_i, chaque occurrence (positive) de $\mathbf{v}$ par $\mathbf{v^i}$ et chaque occurrence de $\neg\mathbf{v}$ par $\mathbf{\bar{v}^i}$; β_i est la formule obtenue. Alors :
   ▷ pour $i = 1, \dots, n$, $\varphi_i = \beta_i \wedge \mathbf{x^i}$,
   ▷ $\varphi_{n+1} = \left(\left(\bigwedge_{\mathbf{v} \in V_\Delta} \left(\bigwedge_{i=1}^n \mathbf{v^i}\right) \vee \left(\bigwedge_{i=1}^n \mathbf{\bar{v}^i}\right)\right) \wedge \mathbf{x^{n+1}}\right) \vee \mathbf{y}$,
   ▷ $\varphi_{n+2} = \mathbf{y}$,
   ▷ $\varphi_{n+3} = \varphi_1$.

Nous allons maintenant prouver la proposition 4.2 à l'aide de plusieurs lemmes.

Lemme 2 *Une allocation $\overrightarrow{\pi}$ pour $\mathcal{P}$ est dite* régulière *si et seulement si pour tout $i \leq n+3$, $\pi_i \subseteq \sigma(i)$, où*
▷ *pour tout $i \leq n$, $\sigma(i) = \bigcup_{\mathbf{v} \in V_\Delta} \{\mathbf{v^i}, \mathbf{\bar{v}^i}\} \bigcup \{\mathbf{x^i}\}$;*
▷ *$\sigma(n+1) = \bigcup_{\mathbf{v} \in V_\Delta, i=1,\dots,n} \{\mathbf{v^i}, \mathbf{\bar{v}^i}\} \bigcup \{\mathbf{x^{n+1}}, \mathbf{y}\}$;*
▷ *$\sigma(n+2) = \{\mathbf{y}\}$.*
▷ *$\sigma(n+3) = \sigma(1)$.*
Étant donnée une allocation $\overrightarrow{\pi}$, soit maintenant $\overrightarrow{\pi}^R$ définie par $\pi_i^R = \pi_i \cap \sigma(i)$. Alors

1. $\overrightarrow{\pi}_R$ *est régulière ;*
2. $\overrightarrow{\pi}$ *est efficace si et seulement si* $\overrightarrow{\pi}^R$ *est efficace ;*
3. *si* $\overrightarrow{\pi}$ *est sans envie alors* $\overrightarrow{\pi}^R$ *est sans envie.*

Démonstration L'affirmation (1) est évidente. Pour tout i, aucun des biens qui ne sont pas dans $\sigma(i)$ n'a d'influence sur la satisfaction de i (puisque ces biens n'apparaissent pas dans φ_i), donc $\pi_i^R \sim_i \pi_i$, d'où l'on peut déduire (2). Les formules φ_i étant positives, les relations de préférence $\succeq_i$ sont monotones, donc on a $\pi_j \succeq_i \pi_j^R$ pour tout (i, j). Maintenant, si $\overrightarrow{\pi}$ est sans envie, alors pour tout (i, j) on a $\pi_i \succeq_i \pi_j$, donc $\pi_i^R \sim_i \pi_i \succeq_i \pi_j \succeq_i \pi_j^R$ et donc $\overrightarrow{\pi}^R$ est sans envie, d'où (3). ▲

Lemme 3 *Si l'allocation* $\overrightarrow{\pi}$ *est régulière alors*

1. 1 *ne peut envier que* $n+3$ *;*
2. $n+3$ *ne peut envier que* 1 *;*
3. $2, \ldots, n$ *n'envient personne ;*
4. $n+1$ *ne peut envier que* $n+2$ *;*
5. $n+2$ *ne peut envier que* $n+1$ *;*

Démonstration Tout d'abord, remarquons que pour tout i, $j \neq i$, i envie j si et seulement si $\pi_i \models \neg\varphi_i$ et $\pi_j \models \varphi_i$.

1. Soient $i = 1$ et $j \in \{2, \ldots, n, n+2\}$. Si 1 envie j, alors $\mathbf{x^1} \in \pi_j$. $\overrightarrow{\pi}$ étant régulière, $\mathbf{x^1} \notin \pi_j$, donc i ne peut envier j.
2. Puisque $\varphi_{n+3} = \varphi_1$, la même affirmation est valable pour $n+3$.
3. Soient $i \in \{2, \ldots, n\}$ et $j \neq i$. Si i envie j alors $\pi_j \models \beta_i \wedge \mathbf{x^i}$, ce qui est impossible car $\mathbf{x^i} \notin \pi_j$, à cause de la régularité de π.
4. Supposons que $n+1$ envie j pour $j \in \{1, \ldots, n, n+3\}$. Alors $\pi_j \models \varphi_{n+1}$. Puisque $\pi_j \models y$ est impossible (car $\overrightarrow{\pi}$ est régulière), on a $\pi_j \models \Big(\bigwedge_{\mathbf{v} \in V_\Delta} \big(\bigwedge_{i=1}^n \mathbf{v^i}\big) \vee \big(\bigwedge_{i=1}^n \bar{\mathbf{v}}^{\mathbf{i}}\big)\Big) \wedge \mathbf{x^{n+1}}$, donc $\pi_j \models \mathbf{x^{n+1}}$, ce qui est impossible aussi, puisque $\overrightarrow{\pi}$ est régulière.
5. Soient $i = n+2$ et $j \in \{1, \ldots, n, n+3\}$. Si i envie j alors $\pi_j \models \mathbf{y}$, ce qui est impossible puisque $\overrightarrow{\pi}$ est régulière. ▲

Lemme 4 *Soit* $\overrightarrow{\pi}$ *une allocation régulière satisfaisant* $n+1$ *et* $n+2$ *et laissant* 1 *et* $n+3$ *non satisfaits. Soit* $M(\overrightarrow{\pi})$ *l'interprétation sur* V_Δ *obtenue à partir de* $\overrightarrow{\pi}$ *par la transformation suivante : pour tout* $\mathbf{v} \in V_\Delta$, $M(\overrightarrow{\pi}) \models \mathbf{v}$ *(c'est-à-dire* $\mathbf{v} \in M(\overrightarrow{\pi})$*) si* $n+1$ *reçoit* $\bar{\mathbf{v}}^{\mathbf{1}}, \ldots, \bar{\mathbf{v}}^{\mathbf{n}}$*, et* $M(\overrightarrow{\pi}) \models \neg\mathbf{v}$ *sinon, c'est-à-dire si* $n+1$ *reçoit* $\mathbf{v^1}, \ldots, \mathbf{v^n}$*. Alors* $\overrightarrow{\pi}$ *est efficace et sans envie seulement si* $M(\pi) \not\models \alpha_1$*.*

Démonstration Soit $\overrightarrow{\pi}$ une allocation régulière satisfaisant $n+1$ et $n+2$. Puisque $\overrightarrow{\pi}$ satisfait $n+2$, $\mathbf{y} \in \pi_{n+2}$. De plus, $\overrightarrow{\pi}$ satisfait $n+1$ sans lui attribuer $\mathbf{y}$, en conséquence, pour tout $\mathbf{v}$, $n+1$ reçoit soit tous les $\mathbf{v^i}$ soit tous les $\bar{\mathbf{v}}^{\mathbf{i}}$. Cela prouve que notre définition de $M(\overrightarrow{\pi})$ est bien fondée.

Maintenant, supposons que $\overrightarrow{\pi}$ est efficace et sans envie, et supposons que $M(\pi) \models \alpha_1$. Il est possible de satisfaire l'un des deux agents 1 et $n+3$ sans qu'aucun agent $j \notin \{1, n+3\}$ ne soit lésé, et ce en lui donnant $\{\mathbf{x^1}\} \cup \bigcup(\bar{\mathbf{v}}^{\mathbf{i}} \mid M(\overrightarrow{\pi}) \models \neg\mathbf{v^i}) \cup \bigcup(\mathbf{v^i} \mid M(\overrightarrow{\pi}) \models \mathbf{v^i})$. Alors, puisque $\overrightarrow{\pi}$ est efficace, cette allocation doit satisfaire au moins l'un des deux agents 1 et $n+3$. Elle ne peut pas les satisfaire tous deux simultanément (à cause de $\mathbf{x^1}$). Ainsi, seul l'un de ces deux agents est satisfait par $\overrightarrow{\pi}$, provoquant donc l'envie de l'autre. Cela contredit l'hypothèse d'absence d'envie de $\overrightarrow{\pi}$, ce qui prouve au final que $M(\overrightarrow{\pi}) \not\models \alpha_1$. ▲

Lemme 5 *Pour toute interprétation M sur V_Δ, définissons $\overrightarrow{\pi}^M \in \wp(\mathscr{O})^n$ par :*
▷ *pour tout $i \in 1,\ldots,n$, $\pi_i^M = \{\mathbf{v^i} \mid M \vDash \mathbf{v}\} \cup \{\bar{\mathbf{v}}^{\mathbf{i}} \mid M \vDash \neg\mathbf{v}\} \cup \{\mathbf{x^i}\}$;*
▷ *$\pi_{n+1}^M = \{\mathbf{x^{n+1}}\} \cup \{\bar{\mathbf{v}}^{\mathbf{i}} \mid M \vDash \mathbf{v}, i = 1,\ldots,n\} \cup \{\mathbf{v^i} \mid M \vDash \neg\mathbf{v}, i = 1,\ldots,n\}$;*
▷ *$\pi_{n+2}^M = \{\mathbf{y}\}$*
▷ *$\pi_{n+3}^M = \varnothing$.*
Alors :

1. *$\overrightarrow{\pi}^M$ est une allocation bien définie et régulière qui satisfait $n+1$ et $n+2$;*
2. *$M(\overrightarrow{\pi}^M) = M$ ($M(\overrightarrow{\pi}^M)$ est obtenu à partir de $\overrightarrow{\pi}^M$ de la même manière que dans le lemme 4) ;*
3. *pour tout $i \in 1,\ldots,n$, $\overrightarrow{\pi}^M$ satisfait i si et seulement si $M \vDash \alpha_i$;*
4. *$\overrightarrow{\pi}^M$ est efficace si et seulement si M satisfait un sous-ensemble maximal consistant de Δ.*

Démonstration 1. On peut aisément vérifier que $\overrightarrow{\pi}^M$ n'attribue pas le même objet à plus d'un seul individu, et que $\overrightarrow{\pi}^M$ peut seulement donner à chaque agent i un ensemble d'objets inclus dans $\sigma(i)$. Ainsi, c'est une allocation bien définie et régulière. Cette allocation satisfait de manière évidente $n+1$ et $n+2$.

2. Si $M \vDash \mathbf{v}$ alors π_{n+1}^M contient $\{\bar{\mathbf{v}}^{\mathbf{i}} \mid i = 1,\ldots,n\}$ et donc $M(\overrightarrow{\pi}^M) \vDash \mathbf{v}$. Le cas $M \vDash \neg v$ est complètement similaire.
3. Soit $i \in 1,\ldots,n$. Puisque $\overrightarrow{\pi}^M$ donne $\mathbf{x^i}$ à l'agent i, $\overrightarrow{\pi}^M$ satisfait i si et seulement si $F(\pi_i^M) \vDash \beta_i$, ce qui est équivalent à $M \vDash \alpha_i$.
4. On peut déduire du point 3 le fait que $\{i \mid \overrightarrow{\pi}^M \text{ satisfait } i\} = \{i \mid M \vDash \alpha_i\} \cup \{n+1, n+2\}$ (de manière évidente, $n+3$ n'est pas satisfaite). De plus, puisque les préférences sont dichotomiques, un partage $\overrightarrow{\pi}$ est efficace si et seulement si l'ensemble des individus qu'il satisfait est maximal pour l'inclusion. Ainsi, $\overrightarrow{\pi}^M$ est efficace si et seulement si M satisfait un sous-ensemble maximal consistant de Δ. ▲

Lemme 6 *Soit $\overrightarrow{\pi}$ une allocation régulière et efficace qui satisfait $n+1$ et $n+2$. Alors $M(\overrightarrow{\pi})$ satisfait un sous-ensemble maximal consistant de Δ.*

Démonstration L'allocation $\overrightarrow{\pi}$ est régulière et satisfait $n+1$ et $n+2$, donc de manière évidente $\pi_{n+2} = \{y\}$ et d'après le lemme 4 $M(\overrightarrow{\pi})$ est bien définie. Nous allons maintenant considérer l'allocation $\overrightarrow{\pi}^{M(\overrightarrow{\pi})}$, définie à partir de $\overrightarrow{\pi}$ de la même manière que dans les lemmes précédents. Nous avons $\pi_{n+1}^{M(\overrightarrow{\pi})} = \{\mathbf{x^{n+1}}\} \cup \{\mathbf{v^i} \mid M(\overrightarrow{\pi}) \vDash \neg\mathbf{v}\} \cup \{\bar{\mathbf{v}}^{\mathbf{i}} \mid M(\overrightarrow{\pi}) \vDash \mathbf{v}\} = \{\mathbf{x^{n+1}}\} \cup \{\mathbf{v^i} \mid \{\mathbf{v^1},\ldots,\mathbf{v^n}\} \subset \pi_{n+1}\} \cup \{\bar{\mathbf{v}}^{\mathbf{i}} \mid \{\bar{\mathbf{v}}^{\mathbf{1}},\ldots,\bar{\mathbf{v}}^{\mathbf{n}}\} \subset \pi_{n+1}\}$. Puisque $n+1$ est satisfait par $\overrightarrow{\pi}$, π_{n+1} doit contenir $\{\mathbf{x^{n+1}}\}$, d'où nous pouvons affirmer que $\pi_{n+1}^{M(\overrightarrow{\pi})} \subseteq \pi_{n+1}$.

Soit $i \in \{2,\ldots,n\}$. Puisque l'allocation $\overrightarrow{\pi}$ est régulière, nous avons $\pi_i \subset \sigma(i)$. Puisque $\overrightarrow{\pi}^{M(\overrightarrow{\pi})}$ est une allocation complète par définition, et régulière d'après le lemme 5, $\sigma(i) \subset \pi_i^{M(\overrightarrow{\pi})} \cup \pi_{n+1}^{M(\overrightarrow{\pi})}$. Puisque $\pi_{n+1}^{M(\overrightarrow{\pi})} \subseteq \pi_{n+1}$ nous avons $\sigma(i) \subset \pi_i^{M(\overrightarrow{\pi})} \cup \pi_{n+1}$, et donc $\pi_i \subset \pi_i^{M(\overrightarrow{\pi})} \cup \pi_{n+1}$. $\overrightarrow{\pi}$ étant une allocation, nous avons bien entendu $\pi_i \cap \pi_{n+1} = \varnothing$, d'où $\pi_i \subseteq \pi_i^{M(\overrightarrow{\pi})}$.

$\overrightarrow{\pi}$ étant régulière, nous avons $\pi_1 \cup \pi_{n+3} \subseteq \sigma(1) \cup \sigma(n+3)$. Puisque $\sigma(1) = \sigma(n+3)$, cette dernière inclusion se ramène à $\pi_1 \cup \pi_{n+3} \subseteq \sigma(1)$. De plus, nous avons $\sigma(1) \subset \pi_1^{M(\overrightarrow{\pi})} \cup \pi_{n+1}^{M(\overrightarrow{\pi})} \cup \pi_{n+3}^{M(\overrightarrow{\pi})}$ pour des raisons similaires à celles que nous avons évoquées pour $i \in \{2,\ldots,n\}$, ce qui se ramène, grâce à $\pi_{n+1}^{M(\overrightarrow{\pi})} \subseteq \pi_{n+1}$ et $\pi_{n+3}^{M(\overrightarrow{\pi})} = \varnothing$, à $\sigma(1) \subset \pi_1^{M(\overrightarrow{\pi})} \cup \pi_{n+1}$, et, à l'aide de $\pi_1 \cap \pi_{n+1} = \varnothing$, nous déduisons l'inclusion $\pi_1 \cup \pi_{n+3} \subseteq \pi_1^{M(\overrightarrow{\pi})}$.

Nous pouvons désormais prouver que $\overrightarrow{\pi}^{M(\overrightarrow{\pi})}$ est efficace. Puisque les préférences sont monotones, tous les individus sauf $n+3$ satisfaits par $\overrightarrow{\pi}$ sont satisfaits aussi par $\overrightarrow{\pi}^{M(\overrightarrow{\pi})}$ (puisque $\forall i \neq n+3$, $\pi_i \subseteq \pi_i^{M(\overrightarrow{\pi})}$).

- ▷ Si $n+3$ n'était pas satisfait par $\overrightarrow{\pi}$, alors nous pouvons immédiatement déduire que $\pi_{M(\overrightarrow{\pi})}$ est efficace.
- ▷ Si $n+3$ était satisfait par $\overrightarrow{\pi}$, supposons que $\overrightarrow{\pi}^{M(\overrightarrow{\pi})}$ n'est pas efficace. Dans ce cas, il existe une allocation $\overrightarrow{\pi}'$ telle que $\forall i$, $\overrightarrow{\pi}^{M(\overrightarrow{\pi})}$ satisfait i implique que $\overrightarrow{\pi}'$ satisfait i et il existe un j particulier ($j \neq 1$) tel que $\overrightarrow{\pi}'$ satisfait j et $\overrightarrow{\pi}^{M(\overrightarrow{\pi})}$ ne satisfait pas j. Clairement, $\overrightarrow{\pi}'$ satisfait 1 (puisque c'est le cas pour $\overrightarrow{\pi}^{M(\overrightarrow{\pi})}$), donc $j \neq n+3$ (car satisfaire simultanément 1 et $n+3$ est impossible). Considérons l'allocation $\overrightarrow{\pi}''$ déduite de $\overrightarrow{\pi}'$ par simple échange des parts de 1 et $n+3$. Nous avons, pour tout $i \in \{2, \ldots, n+2\}$, $\overrightarrow{\pi}$ satisfait i implique $\pi^{M(\overrightarrow{\pi})}$ satisfait i implique à son tour $\overrightarrow{\pi}'$ satisfait i implique enfin $\overrightarrow{\pi}''$ satisfait i. Nous avons aussi $\overrightarrow{\pi}$ satisfait $n+3$ et ne satisfait pas 1, ce qui est la même chose pour $\overrightarrow{\pi}''$. Ainsi $\overrightarrow{\pi}$ satisfait i implique $\overrightarrow{\pi}''$ satisfait i pour tout i. De plus, $\overrightarrow{\pi}''$ satisfait $j \in \{2, \ldots, n+2\}$ (le même j que ci-avant) alors que ce n'est pas le cas pour $\overrightarrow{\pi}^{M(\overrightarrow{\pi})}$, et donc ce n'est pas le cas non plus pour $\overrightarrow{\pi}$. Cela prouve que $\overrightarrow{\pi}$ est Pareto-dominé, ce qui est contradictoire avec les hypothèses.

En conséquence, $\pi^{M(\overrightarrow{\pi})}$ est efficace, d'où nous pouvons conclure avec le lemme 5 (point 4), que $M(\overrightarrow{\pi})$ satisfait un sous-ensemble maximal consistant de Δ. ▲

Lemme 7 *Toute allocation efficace et sans envie pour $\mathcal{P}(\Delta)$ satisfait $n+1$ et $n+2$, et laisse 1 et $n+3$ insatisfaits.*

Démonstration Supposons que $\overrightarrow{\pi}$ ne satisfait pas $n+1$; alors $\mathbf{y} \notin \pi_{n+1}$. Maintenant, si $\mathbf{y} \in \pi_{n+2}$ alors $n+1$ envie $n+2$. Si $\mathbf{y} \notin \pi_{n+2}$ alors $\overrightarrow{\pi}$ n'est pas efficace, car si l'on donnait $\mathbf{y}$ à $n+2$ on le satisferait et cela conduirait à une meilleure allocation que $\overrightarrow{\pi}$.

Maintenant, supposons que $\overrightarrow{\pi}$ ne satisfait pas $n+2$, c'est-à-dire que $\mathbf{y} \notin \pi_{n+2}$. Si $\mathbf{y} \in \pi_{n+1}$ alors $n+2$ envie $n+1$. Si $\mathbf{y} \notin \pi_{n+1}$ alors encore une fois $\overrightarrow{\pi}$ n'est pas efficace, car si l'on donnait $\mathbf{y}$ à $n+1$ on le satisferait et cela conduirait à une meilleure allocation que $\overrightarrow{\pi}$.

En ce qui concerne les agents 1 et $n+3$, on peut remarquer que puisqu'ils ont des préférences identiques, toute allocation sans envie doit soit les satisfaire tous les deux, soit les laisser tous deux insatisfaits. Puisqu'ils ne peuvent être simultanément satisfaits (à cause de $\mathbf{x^1}$), tout partage sans envie doit les laisser tous deux insatisfaits. ▲

Lemme 8 *S'il existe une allocation EEF, alors il existe un sous-ensemble maximal consistant de Δ qui ne contient pas α_1.*

Démonstration Soit $\overrightarrow{\pi}$ une allocation efficace et sans envie. D'après le lemme 2, $\overrightarrow{\pi}^R$ est régulière, efficace et sans envie. D'après le lemme 7, $\overrightarrow{\pi}^R$ satisfait $n+1$ et $n+2$ et laisse 1 et $n+3$ insatisfaits. Alors d'après le lemme 6, $M(\overrightarrow{\pi}^R)$ satisfait un sous-ensemble maximal consistent de Δ, et d'après le lemme 4, $M(\overrightarrow{\pi}^R) \not\models \alpha_1$. Ainsi $\{\alpha_i \in \Delta \mid M(\overrightarrow{\pi}^R) \models \alpha_i\}$ est un sous-ensemble maximal consistant de Δ et ne contient pas α_1. ▲

Lemme 9 *S'il existe un sous-ensemble maximal consistant de Δ qui ne contient pas α_1 alors il existe une allocation EEF.*

Démonstration Supposons qu'il existe un sous-ensemble $\mathscr{S}$ maximal consistant de Δ qui ne contient pas α_1, et soit M un modèle de $\bigwedge_{\varphi \in \mathscr{S}} \varphi$. D'après le point 4 du lemme 5, $\overrightarrow{\pi}^M$ est efficace.

D'après le point 1 du lemme 5, l'allocation $\overrightarrow{\pi}^M$ est régulière ; donc d'après le lemme 3, $\overrightarrow{\pi}^M$ est sans envie si et seulement si (i) 1 n'envie pas $n+3$, (ii) $n+3$ n'envie pas 1 (iii) $n+1$ n'envie pas $n+2$ et (iv) $n+2$ n'envie pas $n+1$. Par définition de $\overrightarrow{\pi}^M$, $\overrightarrow{\pi}^M$ ne satisfait pas $n+3$, donc (i) est vérifié. D'après le point 3 du lemme 5, $M \not\models \alpha_1$ implique que $\overrightarrow{\pi}^M$ ne satisfait pas 1, en conséquence (ii) est aussi vérifié. Finalement, d'après le point 1 du lemme 4, $n+1$ et $n+2$ sont satisfaits par $\overrightarrow{\pi}^M$, et donc (iii) et (iv) sont vérifiés. Ainsi, $\overrightarrow{\pi}^M$ est sans envie. ▲

Nous avons désormais réuni tout le matériel nécessaire pour prouver la proposition 4.2 :

Démonstration (Proposition 4.2) D'après les lemmes 8 et 9, l'existence d'un sous-ensemble maximal consistant de Δ qui ne contient pas α_1 et l'existence d'un partage efficace et sans envie pour $\mathcal{P}(\Delta)$ sont équivalents. Clairement, $\mathcal{P}(\Delta)$ est calculée en temps polynomial. Ainsi, $\mathcal{P}$ est une réduction polynomiale de $\overline{[\mathsf{RSI}]}$ vers [EEF EXISTENCE], ce qui montre la Σ_2^{p}-difficulté de ce dernier problème, et finalement sa Σ_2^{p}-complétude. ▲

Un corollaire évident à cette proposition est que le résultat de Σ_2^{p}-complétude est valable pour des préférences dichotomiques générales (non nécessairement monotones) :

Corollaire 1 *Le problème [EEF EXISTENCE] pour des préférences dichotomiques générales sous forme logique est Σ_2^{p}-complet.*

4.1.1.2 Restrictions sur le langage

La complexité élevée du problème général a pour conséquence qu'il peut être intéressant de s'intéresser aux restrictions et aux variantes de ce problème pour lesquelles cette complexité peut décroître. Nous allons analyser trois types de restrictions intuitives du problème [EEF EXISTENCE], définies respectivement :

- ▷ en fixant le nombre d'agents, et plus précisément en restreignant le problème au cas où il n'y a que deux agents ;
- ▷ en imposant des préférences identiques pour tous les agents ;
- ▷ en restreignant la syntaxe des buts des agents, en limitant leur expression à certaines sous-classes de formules propositionnelles (par exemple les clauses, les cubes, ...).

Contrairement au problème général [EEF EXISTENCE], la complexité de ces restrictions est potentiellement sensible au fait que les préférences soient monotones ou non.

Préférences identiques Considérons tout d'abord le cas pour lequel les agents ont des préférences dichotomiques identiques, c'est-à-dire que toutes les formules φ_i sont identiques.

Proposition 4.3 *Le problème [EEF EXISTENCE] avec n préférences identiques monotones dichotomiques est* NP*-complet. Ce résultat reste valable pour un nombre fixé d'agents $n \geq 2$.*

Démonstration Si les préférences sont identiques, tout partage sans envie doit satisfaire soit tous les agents, soit aucun d'entre eux. Maintenant, si les préférences sont monotones, il est toujours possible de satisfaire au moins un agent (en lui donnant tous les objets). En conséquence, une allocation est EEF si et seulement si elle satisfait tous les agents. On peut clairement vérifier en temps polynomial qu'une allocation donnée satisfait tous les agents, d'où l'appartenance à NP.

La difficulté vient d'une simple réduction depuis le problème [SET SPLITTING] :

Problème 3: [SET SPLITTING]	
INSTANCE :	Une collection $C = \{\mathscr{C}_1, \ldots, \mathscr{C}_n\}$ de sous-ensembles d'un ensemble fini $\mathscr{S}$.
QUESTION :	Existe-t-il une partition $\langle \mathscr{S}_1, \mathscr{S}_2 \rangle$ de $\mathscr{S}$ telle qu'aucun des sous-ensembles de C n'est entièrement contenu ni dans $\mathscr{S}_1$ ni dans $\mathscr{S}_2$?

Soit $(C, \mathscr{S})$ une instance de [SET SPLITTING], et soit $\mathcal{P}(C, \mathscr{S})$ l'instance suivante de [EEF EXISTENCE] :

Agents : 2 agents,

Objets : un objet $o(a)$ par élément $a \in \mathscr{S}$,

Préférences : $\varphi_1 = \varphi_2 = \bigwedge_{\mathscr{C}_i \in C} \bigvee_{a \in \mathscr{C}_i} \mathbf{o}(a)$ (et comme à l'accoutumée $\varphi_k^* = \bigwedge_{\mathscr{C}_i \in C} \bigvee_{a \in \mathscr{C}_i} \mathbf{alloc}(\mathbf{o}(\mathbf{a}), \mathbf{k})$) : chaque agent désire au moins un objet de chaque ensemble.

Il est facile de voir que s'il existe au moins une partition $\langle \mathscr{S}_1, \mathscr{S}_2 \rangle$ de $(C, \mathscr{S})$ qui vérifie les conditions du problème [SET SPLITTING], il est possible de trouver une allocation qui satisfait les deux agents, en leur donnant respectivement $o(\mathscr{S}_1)$ et $o(\mathscr{S}_2)$. Réciproquement, supposons qu'il existe une allocation efficace et sans envie $\overrightarrow{\pi}$, alors cette allocation doit satisfaire les deux agents. Soit $\langle \mathscr{S}_1, \mathscr{S}_2 \rangle = \langle o^{-1}(\pi_1), o^{-1}(\pi_2) \rangle$. Supposons qu'il existe un $\mathscr{C}_i \in C$ tel que $\mathscr{C}_i \subset \mathscr{S}_1$ ou $\mathscr{C}_i \subset \mathscr{S}_2$ (disons par exemple $\mathscr{C}_i \subset \mathscr{S}_1$). Alors $\bigvee_{a \in \mathscr{C}_i} \mathbf{alloc}(\mathbf{o}(\mathbf{a}), \mathbf{2})$ est faux, ce qui falsifie φ_2^*, ce qui est contradictoire avec l'hypothèse initiale. En conséquence $\langle \mathscr{S}_1, \mathscr{S}_2 \rangle$ est une partition de $(C, \mathscr{S})$ qui vérifie les conditions du problème [SET SPLITTING].

Cette réduction est clairement polynomiale, d'où la NP-difficulté du problème [EEF EXISTENCE] avec 2 agents ayant des préférences dichotomiques identiques et monotones. ▲

Contrairement à la proposition 4.2, la proposition 4.4 est sensible à la monotonie des préférences.

Proposition 4.4 *Le problème [EEF EXISTENCE] avec n préférences identiques dichotomiques est co-BH_2-complet. Ce résultat reste valable pour un nombre fixé d'agents $n \geq 2$.*

Démonstration Si les préférences sont identiques, tout partage sans envie doit satisfaire soit tous les agents, soit aucun d'entre eux. Maintenant, soit φ la formule représentant les préférences d'un agent (bien entendu φ est identique pour tous les agents). Si φ est satisfiable alors il est possible de satisfaire au moins un agent. Dans ce cas, une allocation $\overrightarrow{\pi}$ est EEF si et seulement si $\overrightarrow{\pi}$ satisfait tous les agents. Si φ n'est pas satisfiable, alors toute allocation est EEF. En conséquence, il existe un partage EEF si et seulement si $\Gamma \wedge \varphi_1^* \ldots\ldots \varphi_n^*$ est satisfiable ou φ ne l'est pas. Cela prouve l'appartenance à co-BH_2.

La difficulté est montrée par une simple réduction depuis [SAT-OR-UNSAT]. Soit (φ, ψ) une paire de formules propositionnelles qui sont supposées (sans perte de généralité) n'avoir aucune variable en commun. Nous pouvons transformer cette paire de formules en l'instance du problème [EEF EXISTENCE] définie comme suit :

Agents : 2 agents ;

Objets : 2 objets o et o' par variable propositionelle $\mathbf{o}$ apparaissant dans φ, un objet p par variable propositionnelle $\mathbf{p}$ apparaissant dans ψ, et un objet y ;

Préférences : $\varphi_1 = \varphi_2 = \varphi \vee \varphi' \vee (y \wedge \psi)$, où φ' désigne la formule φ dans laquelle chaque variable $\mathbf{o}$ a été remplacée par $\mathbf{o}'$.

1. Supposons que φ n'est pas satisfiable, mais que ψ l'est (ce qui correspond à une instance négative de [SAT-OR-UNSAT]). Alors il est possible de satisfaire au moins un agent en lui donnant y et les objets o correspondant aux variables instanciées à vrai dans le modèle de ψ. Cependant, il n'est pas possible de satisfaire le deuxième agent simultanément, car φ n'est pas satisfiable (et en conséquence φ' ne l'est pas non plus), et le premier agent a déjà pris y. En conséquence, il n'y a pas de partage EEF dans ce cas.

2. Supposons maintenant que φ est satisfiable ou bien ψ ne l'est pas (ce qui correspond à une instance positive de [SAT-OR-UNSAT]). Il y a deux cas :
   - ▷ φ est satisfiable. Dans ce cas, peu importe que ψ soit satisfiable ou pas, il est possible de satisfaire les deux agents en satisfaisant simultanément φ pour le premier d'entre eux et φ' pour le second. En conséquence il y a une allocation EEF.
   - ▷ φ et ψ sont tous deux non satisfiables (rappelons que le cas φ insatisfiable et ψ satisfiable est déjà couvert par le point 1). Dans ce cas, il est clairement impossible de satisfaire un agent, et donc l'allocation vide est efficace et sans envie.

En conséquence, il existe une allocation EEF si et seulement si φ est satisfiable ou ψ ne l'est pas, ce qui prouve la proposition. ▲

Deux agents Nous pouvons constater que pour les deux résultats précédents, la difficulté du problème subsiste même si le nombre d'agents est *fixé* (supérieur à 2). Les choses sont différentes avec la proposition 4.2, pour laquelle la difficulté chute lorsque l'on fixe le nombre d'agents. Littéralement, nous avons le résultat suivant :

Proposition 4.5 *Le problème [EEF EXISTENCE] pour deux agents avec des préférences monotones dichotomiques est* NP*-complet.*

Démonstration La NP-difficulté est un corollaire de la proposition 4.4. L'appartenance à NP est obtenue comme suit. Soit (φ_1, φ_2) la paire de formules représentant les préférences des agents, où φ_1, φ_2 sont toutes deux positives. Les formules Γ, Λ, ainsi que les formules φ_i^* sont définies comme précédemment (voir section 3.3.1), de même que $F(\overrightarrow{\pi})$ pour toute allocation $\overrightarrow{\pi}$. L'allocation $\overrightarrow{\pi}$ est efficace si et seulement si soit (a) elle satisfait les deux agents, soit (b) elle ne satisfait qu'un seul des deux agents, et $\Gamma \wedge \varphi_1^* \wedge \varphi_2^*$ est insatisfiable, soit (c) il est impossible de satisfaire même un seul agent, c'est-à-dire que φ_1 et φ_2 sont toutes deux insatisfiables. Le cas (c) est impossible car φ_1 et φ_2 sont positives. En outre, $\overrightarrow{\pi}$ est sans envie si et seulement si $F(\overrightarrow{\pi}) \models \Lambda$. En conséquence, $\overrightarrow{\pi}$ est EEF si et seulement si soit (a) $F(\overrightarrow{\pi}) \models \Gamma \wedge \varphi_1^* \wedge \varphi_2^*$, soit (b) $F(\overrightarrow{\pi}) \models \Gamma \wedge (\varphi_1^* \vee \varphi_2^*) \wedge \Lambda$. En conséquence, il existe une allocation EEF si et seulement si $(\Gamma \wedge \varphi_1^* \wedge \varphi_2^*) \vee (\Gamma \wedge (\varphi_1^* \vee \varphi_2^*) \wedge \Lambda)$ est satisfiable, d'où l'appartenance à NP. ▲

Proposition 4.6 *Le problème [EEF EXISTENCE] pour 2 agents avec des préférences dichotomiques est co-*BH_2*-complet.*

Démonstration La preuve d'appartenance est fondée sur la réduction suivante vers [SAT-OR-UNSAT]. Soit $\mathcal{P}$ une instance du problème EEF avec 2 agents ayant respectivement les préférences φ_1 et φ_2. Nous transformons cette instance en une instance (ψ, ψ') de [SAT-OR-UNSAT], définie comme suit (la formule Γ est définie comme à l'accoutumée) : $\psi = (\Gamma \wedge \varphi_1^* \wedge \varphi_2^*) \vee (\varphi_1 \wedge \neg\varphi_2) \vee (\neg\varphi_1 \wedge \varphi_2)$ et $\psi' = (\varphi_1 \vee \varphi_2)$. Nous allons montrer que ψ est satisfiable ou ψ' est insatisfiable si et seulement s'il existe une allocation EEF pour $\mathcal{P}$.

1. Supposons que ψ est non satisfiable, mais que ψ' l'est. Puisque ψ est non satisfiable, aucun partage valide ne peut satisfaire les deux agents (car $\Gamma \wedge \varphi_1^* \wedge \varphi_2^*$ est non satisfiable). Nous pouvons donc déduire que toute allocation efficace satisfait exactement un agent. Puisque $(\varphi_1 \wedge \neg\varphi_2) \vee (\neg\varphi_1 \wedge \varphi_2)$ n'est pas satisfiable, $Mod(\varphi_1) = Mod(\varphi_2)$ (en d'autres termes φ_1 et φ_2 sont logiquement équivalents). Soit $\overrightarrow{\pi}$ l'allocation qui satisfait l'agent 1 (le cas est similaire avec l'agent 2), $F(\overrightarrow{\pi}) \models \varphi_1^*$, et donc $F(\overrightarrow{\pi}) \models \varphi_{2|1}^*$. Puisque $F(\overrightarrow{\pi}) \not\models \varphi_2^*$ (car il est impossible de satisfaire les deux agents), $F(\overrightarrow{\pi}) \not\models \Lambda$ et donc $\overrightarrow{\pi}$ n'est pas sans envie. D'où le fait qu'aucune allocation efficace n'est sans envie : en d'autres termes, il n'existe aucune allocation EEF.

2. Supposons maintenant que ψ est satisfiable ou ψ' ne l'est pas.
   - ▷ ψ' n'est pas satisfiable. Ce cas peut être élucidé facilement, car aucun des deux agents n'est satisfiable. Dans ce cas, toute allocation est efficace et sans envie.
   - ▷ φ et ψ sont toutes deux satisfiables. On peut distinguer deux cas :
     - $\Gamma \wedge \varphi_1 \wedge \varphi_2$ est satisfiable. Dans ce cas, il y a une allocation, correspondant au modèle de $\Gamma \wedge \varphi_1 \wedge \varphi_2$, qui satisfait les deux agents. Cette allocation est clairement EEF.
     - $\varphi_1 \wedge \neg\varphi_2$ est satisfiable, mais $\Gamma \wedge \varphi_1 \wedge \varphi_2$ ne l'est pas (le cas avec $\varphi_2 \wedge \neg\varphi_1$ est similaire). Dans ce cas, il n'est pas possible de satisfaire les deux agents simultanément. Cependant, puisque ψ est satisfiable, il est possible d'en satisfaire au moins un, et, comme dans le point 1, toute allocation efficace satisfait exactement un agent. Puisque $\varphi_1 \wedge \neg\varphi_2$ est satisfiable, il existe un modèle de φ_1 qui n'est pas un modèle de φ_2. L'allocation correspondant à ce modèle est telle que l'agent 1 est satisfait et l'agent 2 ne l'est pas, mais l'agent 2 ne peut pas envier l'agent 1.

Cela montre finalement la correction de la réduction, qui est clairement polynomiale.
La difficulté vient directement de la proposition 4.4. ▲

Restrictions sur le langage propositionnel Dans les résultats précédents, nous n'avons fait aucune hypothèse spécifique sur les formules exprimant les préférences des agents, excepté (parfois) leur positivité, correspondant à la propriété de monotonie. Cependant, si nous restreignons l'ensemble des formules propositionnelles possibles, cela peut éventuellement faire décroître la complexité du problème [EEF EXISTENCE]. Nous allons nous pencher sur deux restrictions naturelles du langage propositionnel : dans le premier cas, nous restreignons l'ensemble des formules à l'ensemble des clauses, et dans le second cas nous limitons les formules propositionnelles à l'ensemble des cubes. Ces restrictions correspondent intuitivement à deux types de problèmes réels.

- ▷ Le cas où les préférences des agents sont représentées par des clauses correspond à un type de problèmes pour lesquels les objets sont regroupés en classes, et chaque agent ne désire qu'un seul objet par classe. On peut prendre l'exemple d'ensemble de patients en attente d'une greffe de rein. Chaque patient ne désire qu'un seul rein, mais parmi les reins disponibles il se peut qu'il y en ait plusieurs qui soient compatibles avec un même agent.
- ▷ Le cas où les préférences des agents sont représentées par des cubes correspond au type de problèmes pour lesquels chaque agent a besoin d'un unique ensemble d'objets. On peut citer par exemple le cas de problèmes pour lesquels les agents construisent l'objet qu'ils désirent à partir d'un ensemble de composants matériels (ou virtuels) basiques : l'ensemble des objets correspond aux composants basique ; et le cube représentant la préférence d'un agent correspond à l'objet construit qu'il désire.

Ces restrictions sur les formules propositionnelles font effectivement chuter la complexité du problème [EEF EXISTENCE]. Dans le cas de la restriction aux clauses d'objets, cela a même pour effet de rendre le problème polynomial.

Proposition 4.7 *Le problème [EEF EXISTENCE] pour des agents ayant des préférences dichotomiques restreintes aux clauses d'objets peut être résolu en temps polynomial.*

Démonstration Nous allons tout d'abord introduire deux hypothèses supplémentaires, et montrer que la complexité du problème ne décroît pas sous ces deux hypothèses. (1) Nous supposons tout d'abord que les préférences des agents sont monotones. Si l'un des agents a des préférences non monotones, cela signifie qu'il y a un littéral négatif dans sa clause. En conséquence, si on lui donne une part vide, on le satisfait sans léser un autre agent. Un tel agent peut donc être retiré du problème sans modifier la complexité. (2) Nous supposons de même que chaque agent désire au moins un objet. Si l'un des agents a une clause vide en tant que but, cela signifie qu'il ne peut être satisfait, quelle que soit la part qu'il reçoit. En conséquence, on peut le retirer du problème sans que cela ne change quoi que ce soit. Dans la suite de la preuve, nous nous limiterons donc aux problèmes qui vérifient les conditions (1) et (2).

La preuve est fondée sur le résultat suivant, que nous allons commencer par démontrer : lorsque les préférences des agents sont des disjonctions d'objets vérifiant les hypothèses (1) et (2), une allocation est Pareto-efficace et sans envie si et seulement si elle satisfait tous les agents. L'implication $\Leftarrow$ est immédiate. Afin de prouver l'implication $\Rightarrow$, il nous faut remarquer qu'un agent est satisfait si et seulement si il reçoit au moins un objet de sa clause. Maintenant, considérons une allocation $\overrightarrow{\pi}$ telle qu'il existe un agent i qui n'est pas satisfait par $\overrightarrow{\pi}$. Alors soit il est possible de le satisfaire sans léser un autre agent, et dans ce cas, $\overrightarrow{\pi}$ n'est pas Pareto-efficace, soit ce n'est pas possible parce que chaque objet de la disjonction de i a été donné à un autre agent qui le désire vraiment. Dans ce dernier cas, l'agent i envie ces autres agents, et $\overrightarrow{\pi}$ n'est donc pas sans envie.

En conséquence, la recherche d'une allocation Pareto-efficace et sans envie se ramène à la recherche d'une allocation qui donne à chaque agent un objet qu'il désire. Ainsi, toute instance $\mathcal{P}$ du problème [EEF EXISTENCE] peut être réduite en une instance du problème de couplage maximal ([MAXIMAL MATCHING], voir problème 11 page 163) dans un graphe bipartite $G_{\mathcal{P}}$ défini comme suit : un nœud par agent d'un côté, un nœud par objet de l'autre, et un arc entre un nœud-agent i et un nœud-objet o si et seulement si **o** apparaît dans la clause de l'agent i. On peut vérifier facilement qu'il existe une allocation Pareto-efficace et sans envie si et seulement s'il existe un couplage de taille n dans $G_{\mathcal{P}}$. Le problème de couplage peut être résolu en temps $O(nm)$ [Ford et Fulkerson, 1962], où m est la taille de la plus grande disjonction, ce qui prouve la proposition. ▲

Maintenant, nous allons nous intéresser au cas où les préférences des agents sont des cubes d'objets. De manière peu surprenante, ce cas est plus difficile que le précédent, sans toutefois dépasser les limites de NP.

Proposition 4.8 *Le problème [EEF EXISTENCE] pour des agents ayant des préférences dichotomiques restreintes aux cubes d'objets est* NP*-complet. Ce résultat reste valable si nous nous restreignons en plus aux préférences monotones.*

Démonstration La preuve est organisée comme suit. Tout d'abord, nous allons prouver l'appartenance à NP sans aucune hypothèse sur la monotonie des préférences. Dans un deuxième temps, nous montrerons la NP-difficulté dans le cas monotone.

Introduisons tout d'abord deux notations supplémentaires. Nous noterons $Obj^+(i)$ (resp. $Obj^-(i)$) l'ensemble des objets apparaissant comme littéraux positifs (resp. négatifs) dans le cube de l'agent i. Soit $\overrightarrow{\pi}$ une allocation. $\overrightarrow{\pi}$ sera dite *minimalement régulière* si pour tout i, soit $\pi_i = Obj^+(i)$, soit $\pi_i = \varnothing$. Pour une allocation donnée $\overrightarrow{\pi}$, nous noterons $\overrightarrow{\pi}^{MR}$ l'allocation minimalement régulière qui lui correspond, c'est-à-dire l'allocation telle que pour tout i, $\pi_i^{MR} = \varnothing$ si $Obj^+(i) \not\subseteq \pi_i$, et $\pi_i^{MR} = Obj^+(i)$ si $Obj^+(i) \subseteq \pi_i$. Nous noterons aussi $Sat(\pi)$ l'ensemble des agents satisfaits par $\overrightarrow{\pi}$: $Sat(\overrightarrow{\pi}^{MR}) = \{i \mid \pi_i^{MR} = Obj^+(i)\}$), et $All(\overrightarrow{\pi}) = \bigcup_{i \in I} \pi_i$ (l'ensemble des objets alloués).

Nous avons le résultat suivant :

Lemme 10 *Soit $\overrightarrow{\pi}$ une allocation. Nous avons :*

▷ *$\overrightarrow{\pi}^{MR}$ est minimalement régulière ;*

▷ *$Sat(\overrightarrow{\pi}) \subseteq Sat(\overrightarrow{\pi}^{MR})$;*
▷ *Si $\overrightarrow{\pi}$ est Pareto-efficace, alors $\overrightarrow{\pi}^{MR}$ est aussi Pareto-efficace.*

Démonstration ▷ Pour tout agent i, $\pi_i^{MR} = \varnothing$ ou $\pi_i^{MR} = Obj^+(i)$ par définition de $\overrightarrow{\pi}^{MR}$. Donc $\overrightarrow{\pi}^{MR}$ est minimalement régulière.

▷ Soit $\overrightarrow{\pi}$ une allocation, et soit i un agent. Si i est satisfait par $\overrightarrow{\pi}$, alors $Obj^+(i) \subseteq \pi_i$ et $Obj^-(i) \cap \pi_i = \varnothing$. Par définition de $\overrightarrow{\pi}^{MR}$, nous avons $\pi_i^{MR} = Obj^+(i)$, et donc nous avons encore $Obj^+(i) \subseteq \pi_i$ et $Obj^-(i) \cap \pi_i = \varnothing$, donc l'agent i reste satisfait par $\overrightarrow{\pi}^{MR}$. Cela prouve que $Sat(\overrightarrow{\pi}) \subseteq Sat(\overrightarrow{\pi}^{MR})$.

▷ Supposons que $\overrightarrow{\pi}$ est Pareto-efficace, et supposons qu'il existe un partage $\overrightarrow{\pi}'$ qui Pareto-domine $\overrightarrow{\pi}^{MR}$. Alors nous avons $Sat(\overrightarrow{\pi}) \subseteq Sat(\overrightarrow{\pi}^{MR}) \subsetneq Sat(\overrightarrow{\pi}')$, ce qui contredit le fait que $\overrightarrow{\pi}$ est Pareto-efficace. Cela prouve le troisième point. ▲

Lemme 11 *Une allocation minimalement régulière $\overrightarrow{\pi}^{MR}$ est Pareto-efficace si et seulement s'il n'existe aucun agent i tel que (a) $i \notin Sat(\overrightarrow{\pi}^{MR})$ et (b) $Obj^+(i) \cap All(\overrightarrow{\pi}^{MR}) = \varnothing$.*

Démonstration Soit $\overrightarrow{\pi}^{MR}$ une allocation minimalement régulière. Supposons qu'il existe un i tel que $i \notin Sat(\overrightarrow{\pi}^{MR})$ et $Obj^+(i) \subseteq \mathscr{O} \setminus All(\overrightarrow{\pi}^{MR})$. Alors l'allocation $\overrightarrow{\pi}'$ telle que $\forall j \neq i$ $\overrightarrow{\pi}'(j) = \overrightarrow{\pi}^{MR}(j)$ et $\overrightarrow{\pi}'(i) = Obj^+(i)$ est bien définie (puisque $Obj^+(i)$ est contenu dans l'ensemble des objets non alloués pour $\overrightarrow{\pi}^{MR}$), et Pareto-domine $\overrightarrow{\pi}^{MR}$ (puisque tous les agents satisfaits par $\overrightarrow{\pi}^{MR}$ sont aussi satisfaits par $\overrightarrow{\pi}'$, et l'agent i est maintenant satisfait par $\overrightarrow{\pi}'$ alors qu'il ne l'était pas par $\overrightarrow{\pi}^{MR}$).

Réciproquement, supposons que $\overrightarrow{\pi}^{MR}$ n'est pas Pareto-efficace, et soit $\overrightarrow{\pi}'$ une allocation Pareto-efficace qui Pareto-domine $\overrightarrow{\pi}^{MR}$. Alors d'après le lemme 10, $\overrightarrow{\pi}'^{MR}$ est Pareto-efficace et Pareto-domine également $\overrightarrow{\pi}^{MR}$. Pour tout $i \in Sat(\overrightarrow{\pi}^{MR})$, $\pi_i^{MR} = \pi_i'^{MR} = Obj^+(i)$ puisque ces allocations sont toutes deux minimalement régulières, et chaque agent satisfait par $\overrightarrow{\pi}^{MR}$ est aussi satisfait par $\overrightarrow{\pi}'^{MR}$. De plus, il existe un $j \notin Sat(\overrightarrow{\pi}^{MR})$ tel que $\pi_j'^{MR} = Obj^+(j)$. Puisque $All(\overrightarrow{\pi}^{MR}) = \bigcup_{i \in Sat(\overrightarrow{\pi}^{MR})} Obj^+(i)$ et puisque $\pi_j'^{MR} \in \mathscr{O} \setminus \bigcup_{i \in Sat(\overrightarrow{\pi}^{MR})} Obj^+(i)$, nous avons $Obj^+(j) \in \mathscr{O} \setminus All(\overrightarrow{\pi}^{MR})$, ce qui prouve au final le lemme. ▲

Les deux lemmes fournissent une procédure pour vérifier si une allocation $\overrightarrow{\pi}$ donnée est Pareto-efficace. Tout d'abord, on calcule $\overrightarrow{\pi}^{MR}$ (ce qui peut être fait en temps polynomial). D'après le lemme 10, $Sat(\overrightarrow{\pi}) \subseteq Sat(\overrightarrow{\pi}^{MR})$. Si l'inclusion est stricte (c'est-à-dire si $Sat(\overrightarrow{\pi}) \subsetneq Sat(\overrightarrow{\pi}^{MR})$), alors $\overrightarrow{\pi}$ n'est de toute évidence pas Pareto-efficace, puisque $\overrightarrow{\pi}^{MR}$ Pareto-domine cette allocation. Sinon, vérifier si $\overrightarrow{\pi}$ est Pareto-efficace revient à vérifier si $\overrightarrow{\pi}^{MR}$ est Pareto-efficace, ce qui revient, selon le lemme 11, à n tests d'inclusion d'ensembles. En outre, vérifier que $\overrightarrow{\pi}$ est sans envie est toujours polynomial. D'où l'on peut déduire que le problème est dans NP.

Nous allons maintenant montrer la difficulté du problème en se concentrant sur les préférences monotones (c'est-à-dire telles que $Obj^-(i) = \varnothing$ pour tout i). La preuve de complétude va nécessiter deux lemmes supplémentaires.

Lemme 12 *Soit $\overrightarrow{\pi}$ une allocation. Supposons que les agents ont des préférences monotones. Si $\overrightarrow{\pi}$ est sans envie, alors $\overrightarrow{\pi}^{MR}$ est sans envie.*

Démonstration Soit $\overrightarrow{\pi}$ une allocation. D'après le lemme 10, $Sat(\overrightarrow{\pi}) \subseteq Sat(\overrightarrow{\pi}^{MR})$. Soit $i \in Sat(\overrightarrow{\pi}^{MR})$. On a $\pi_i^{MR} \subseteq \pi_i$, ce qui montre que $i \in Sat(\pi)$, parce que nous traitons de préférences monotones. Maintenant supposons que l'agent i envie l'agent j dans $\overrightarrow{\pi}^{MR}$. Alors i n'est pas satisfait par $\overrightarrow{\pi}^{MR}$, et donc n'est pas satisfait non plus par $\overrightarrow{\pi}$. Puisque $\pi_j^{MR} \subseteq \pi_j$ et puisque les préférences de l'agent i sont monotones, l'agent i va continuer d'envier j dans $\overrightarrow{\pi}$. En conséquence, si $\overrightarrow{\pi}$ est sans envie, alors $\overrightarrow{\pi}^{MR}$ l'est aussi. ▲

Un corollaire important de ce lemme est que lorsque nous nous restreignons aux cubes monotones, l'existence d'une allocation Pareto-efficace et sans envie est équivalente à l'existence d'une allocation minimalement régulière Pareto-efficace et sans envie. En conséquence, nous pouvons restreindre notre problème d'existence aux allocations minimalement régulières.

Lemme 13 *Soient i et j deux agents différents (nous supposons toujours que les agents ont des préférences monotones). Alors : (il existe une allocation minimalement régulière $\overrightarrow{\pi}^{MR}$ telle que l'agent i envie l'agent j) si et seulement si ($Obj^+(i) \subseteq Obj^+(j)$).*

Démonstration Soit $\overrightarrow{\pi}^{MR}$ une allocation minimalement régulière. Supposons que i envie j. Alors clairement i n'est pas satisfait mais j l'est ; d'où $\overrightarrow{\pi}^{MR}(j) = Obj^+(j)$. Puisque i envie j, nous avons donc directement $Obj^+(i) \subseteq Obj^+(j)$.

Réciproquement, supposons que $Obj^+(i) \subseteq Obj^+(j)$. Alors l'allocation $\overrightarrow{\pi}^{MR}$ qui attribue $Obj^+(j)$ à l'agent j et rien aux autres agents est très clairement minimalement régulière, et elle est aussi clairement telle que i envie j. ▲

Introduisons maintenant le problème NP-complet qui va nous servir de base pour prouver la NP-difficulté (toujours dans le cas monotone) :

Problème 4: [EXACT COVER BY 3-SETS] [Karp, 1972]

INSTANCE :	Un ensemble $\mathscr{S}$ de taille $3q$, et une collection $C = \{\mathscr{C}_1, \dots, \mathscr{C}_{\lvert C \rvert}\}$ de sous-ensembles à 3 éléments de $\mathscr{S}$
QUESTION :	C contient-il une couverture exacte pour $\mathscr{S}$, c'est-à-dire une sous-collection $C' \subseteq C$ telle que chaque élément de $\mathscr{S}$ apparaît dans exactement un membre de C' ?

Soit $(\mathscr{S}, C = \{\mathscr{C}_1, \dots, \mathscr{C}_{\lvert C \rvert}\})$ une instance de [EXACT COVER BY 3-SETS] (nous supposons sans perte de généralité que les $\mathscr{C}_i$ sont tous différents), et soit $\mathcal{P}(\mathscr{S}, C)$ l'instance suivante du problème [EEF EXISTENCE] :

Agents : un ensemble de $\lvert C \rvert + 2\lvert \mathscr{S} \rvert$ agents $\mathscr{N} = \mathscr{N}_1 \cup \mathscr{N}_2$, avec $\mathscr{N}_1 = \{1, \dots, \lvert C \rvert\}$ et $\mathscr{N}_2 = \{\lvert C \rvert + 1, \dots, \lvert C \rvert + 2\lvert \mathscr{S} \rvert\}$;

Objets : un ensemble de $2\lvert \mathscr{S} \rvert$ objets $\mathscr{O} = \mathscr{O} \cup \mathscr{O}'$, avec $\mathscr{O} = \{o_1, \dots, o_{\lvert \mathscr{S} \rvert}\}$ et $\mathscr{O}' = \{o'_1, \dots, o'_{\lvert \mathscr{S} \rvert}\}$, chaque paire (o_i, o'_i) correspondant à un élément différent a_i de $\mathscr{S}$;

Préférences : pour tout agent $i \in \mathscr{N}_1$, $\varphi_i = \bigwedge_{a_j \in \mathscr{S}_i} \mathbf{o_j}$, et pour tout agent $k \in \{1, \dots, \lvert \mathscr{S} \rvert\}$, $\varphi_{\lvert C \rvert + 2k - 1} = \varphi_{\lvert C \rvert + 2k} = \mathbf{o_k} \wedge \mathbf{o'_k}$.

En d'autres termes, les préférences des $\lvert C \rvert$ premiers agents correspondent aux ensembles de la collection C, et les $2\lvert \mathscr{S} \rvert$ derniers agents sont regroupés par paire, chaque membre de la même paire ayant les mêmes préférences que l'autre membre.

Puisque tous les $\mathscr{S}_i$ sont différents et de taille 3, pour tout $i \neq j$, $\mathscr{S}_i \not\subseteq \mathscr{S}_j$, et donc $Obj^+(i) \not\subseteq Obj^+(j)$. Par définition des préférences, nous avons aussi $Obj^+(i) \not\subseteq Obj^+(j)$ pour tout $(i, j) \in \mathscr{N}_1 \times \mathscr{N}_2$ et de même pour tout $(i, j) \in \mathscr{N}_2 \times \mathscr{N}_1$. En conséquence, d'après le lemme 13, la seule source potentielle d'envie dans une telle instance ne peut venir que d'un agent de $\mathscr{N}_2$ qui envie son partenaire. Puisqu'il est impossible de satisfaire les deux agents de la même paire en même temps, un partage est sans envie si et seulement si il ne satisfait aucun agent de $\mathscr{N}_2$.

D'après le lemme 11, une allocation minimalement régulière $\overrightarrow{\pi}^{MR}$ est Pareto-efficace si et seulement s'il n'existe aucun i tel que $i \notin Sat(\overrightarrow{\pi}^{MR})$ et $Obj^+(i) \subseteq \mathscr{O} \setminus All(\overrightarrow{\pi}^{MR})$. En conséquence, une allocation minimalement régulière $\overrightarrow{\pi}^{MR}$ est Pareto-efficace et sans envie si et seulement s'il n'existe aucun $k \in \{1, \ldots, |\mathscr{S}|\}$ tel que $\pi^{MR}_{|C|+2k-1} = \{o_k, o'_k\}$ ou $\pi^{MR}_{|C|+2k} = \{o_k, o'_k\}$, et il n'existe aucun $k' \in \{1, \ldots, |\mathscr{S}|\}$ tel que $\{o_{k'}, o'_{k'}\} \subset \mathscr{O} \setminus All(\overrightarrow{\pi}^{MR})$ (cette dernière condition se ramène à $o_{k'} \notin All(\overrightarrow{\pi}^{MR})$, puisque $o_{k'}$ et $o'_{k'}$ doivent être alloués ensemble dans toute allocation minimalement régulière). Finalement, $\overrightarrow{\pi}^{MR}$ est Pareto-efficace et sans envie si et seulement si $\forall k \in \{1, \ldots, |\mathscr{S}|\}$, il existe un agent $i \in \mathscr{N}_1$ tel que $o_k \in \pi^{MR}_i$, c'est-à-dire si et seulement si $\bigcup_{i \in \mathscr{N}_1} \pi^{MR}_i = \bigcup_{a_j \in \mathscr{S}} \{o_j\}$.

Soit $\overrightarrow{\pi}^{MR}$ une allocation minimalement régulière. Nous pouvons alors définir la sous-collection $\zeta(\overrightarrow{\pi}^{MR})$ de la manière suivante : $\zeta(\overrightarrow{\pi}^{MR}) = \{\mathscr{C}_i \in C \mid \pi^{MR}_i = Obj^+(i)\}$. L'application ζ définit clairement une bijection entre l'ensemble des allocations minimalement régulières et l'ensemble des sous-collections dont les éléments sont deux à deux disjoints, et on peut remarquer de plus que $\bigcup_{i \in \mathscr{N}_1} \pi^{MR}_i = \bigcup_{\mathscr{C}_j \in \zeta(\overrightarrow{\pi}^{MR})} \bigcup_{a_k \in \mathscr{C}_j} \{o_k\})$.

Soit $C' \subseteq C$ une couverture exacte pour $\mathscr{S}$. Alors $\zeta^{-1}(C')$ existe et est une allocation minimalement régulière valide. Nous avons de plus $\bigcup_{i \in \mathscr{N}_1} \zeta^{-1}(C')(i) = \bigcup_{\mathscr{C}_j \in C} \bigcup_{a_k \in \mathscr{C}_j} \{o_k\} = \bigcup_{a_j \in \mathscr{S}} \{o_j\}$ car C' est une couverture. En conséquence, $\zeta^{-1}(C')$ est Pareto-efficace et sans envie d'après le résultat précédent.

Réciproquement, supposons qu'il existe une allocation $\overrightarrow{\pi}^{MR}$ minimalement régulière, Pareto-efficace et sans envie. Alors $\zeta(\overrightarrow{\pi}^{MR})$ est une sous-collection de C dont les éléments sont deux à deux disjoints, et est tel que $\bigcup_{\mathscr{C}_i \in \zeta(\overrightarrow{\pi}^{MR})} \bigcup_{a_j \in \mathscr{C}_i} \{a_j\} = \bigcup_{i \in \mathscr{N}_1} \bigcup_{o_j \in \pi^{MR}_i} \{a_j\} = \bigcup_{o_j \in \{\pi^{MR}_i \mid i \in \mathscr{N}_1\}} \{a_j\}$. $\overrightarrow{\pi}^{MR}$ étant Pareto-efficace et sans envie, nous avons $\bigcup_{i \in \mathscr{N}_1} \pi^{MR}_i = \bigcup_{a_j \in \mathscr{S}} \{o_j\}$, et donc $\bigcup_{o_j \in \{\pi^{MR}_i \mid i \in \mathscr{N}_1\}} \{a_j\} = \bigcup_{o_j \in \{o_j \mid a_j \in \mathscr{S}\}} \{a_j\} = \mathscr{S}$. Cela prouve que $\zeta(\overrightarrow{\pi}^{MR})$ est une couverture exacte pour $\mathscr{S}$.

Cette réduction est clairement polynomiale, d'où la NP-difficulté. ▲

La preuve précédente (et particulièrement le lemme 13) met en évidence le nœud du problème avec préférences conjonctives, c'est-à-dire le point qui concentre toute la difficulté de ce problème. Dans une instance de ce problème, la seule source d'envie potentielle vient de $Obj^+(i) \subseteq Obj^+(j)$ (si toutefois nous nous restreignons aux allocations minimalement régulières). Dans ce cas, nous ne pouvons pas satisfaire l'agent j sans créer de l'envie de la part de i pour cet agent. Maintenant, si $Obj^+(i) \subsetneq Obj^+(j)$, on peut enlever l'agent j de l'instance, car s'il est satisfait, alors forcément i va l'envier (remarquons toutefois que ce n'est vrai que pour des préférences monotones, car dans le cas contraire on peut donner à un agent un objet qui n'apparaît pas dans ses préférences, dans le seul but d'empêcher un autre agent de l'envier, et donc nous ne pouvons pas nous restreindre à des allocations minimalement régulières).

S'il n'y a aucune paire d'agents (i, j), $i \neq j$, telle que $Obj^+(i) = Obj^+(j)$, alors on peut enlever du problème tous les agents i tels qu'il existe un autre agent j tel que $Obj^+(j) \subsetneq Obj^+(i)$. Il est facile de voir qu'après cette opération, toute allocation minimalement régulière est sans envie. Puisqu'il y a au moins une allocation minimalement régulière Pareto-efficace, cela garantit l'existence d'un partage Pareto-efficace et sans envie dans ce cas.

Plus formellement, on a :

Proposition 4.9 *Il existe toujours un partage efficace et sans envie pour une instance du problème [EEF EXISTENCE] avec des agents ayant des préférences dichotomiques restreintes à des cubes d'objets lorsque la condition suivante est vérifiée :*

$$\forall (i, j) \in \mathscr{N}^2,\ i \neq j,\ (\varphi_i = \varphi_j) \Rightarrow \big(\exists k \text{ tel que } k \neq i,\ k \neq j \text{ et } Obj^+(k) \subsetneq Obj^+(i)\big)\,. \tag{4.1}$$

Bien entendu, il n'y a pas équivalence entre la condition 4.1 et l'existence d'un partage Pareto-efficace et sans envie (sinon la proposition 4.8 serait fausse[1]), car il peut arriver qu'étant donnés deux agents i et j ayant les mêmes préférences, la satisfaction de l'un de ces agents soit empêchée par un autre agent k tel que $Obj^+(i) \cap Obj^+(k) \neq \varnothing$ mais $Obj^+(k) \not\subset Obj^+(i)$. Voilà le cas difficile : lorsque deux agents i et j ont des préférences identiques, mais qu'aucun agent k n'a de préférences telles que $Obj^+(k) \subsetneq Obj^+(i)$, il peut cependant être possible d'empêcher i et j d'être satisfaits, à l'aide d'un autre agent, comme le montre l'exemple suivant : $\varphi_1 = \varphi_2 = \mathbf{o_1} \wedge \mathbf{o_2}$, et $\varphi_2 = \mathbf{o_2} \wedge \mathbf{o_3}$. Satisfaire l'agent 2 uniquement conduit à une allocation efficace et sans envie, alors que la condition 4.1 n'est pas vérifiée.

Démonstration (Proposition 4.9) Dans la preuve, nous noterons $\mathscr{N}_1$ l'ensemble des agents dont les préférences sont «minimales pour l'inclusion», c'est-à-dire que $\mathscr{N}_1 = \{i \mid \nexists j \text{ tel que } Obj^+(j) \subset Obj^+(i)\}$. Nous noterons $\mathscr{N}_2$ l'ensemble des autres agents : $\mathscr{N}_2 = \mathscr{N} \setminus \mathscr{N}_1$.

Voici une procédure simple pour trouver une allocation Pareto-efficace et sans envie : sélectionner de manière gloutonne un ensemble $\mathscr{S}$ maximal d'agents de $\mathscr{N}_1$, tel que chaque agent $i \in \mathscr{S}$ reçoit $Obj^+(i)$ (jusqu'à ce qu'il devienne impossible de sélectionner un nouvel agent dans $\mathscr{N}_1$).

L'allocation $\overrightarrow{\pi}$ qui résulte de cette procédure est minimalement régulière, et d'après le lemme 13 elle est clairement sans envie (par définition de $\mathscr{N}_1$). De plus, supposons qu'il existe un $i \notin Sat(\overrightarrow{\pi})$ tel que $Obj^+(i) \subseteq \mathscr{O} \setminus All(\pi)$. Alors $i \notin \mathscr{N}_1$, puisque si c'était le cas, la procédure aurait sélectionné cet agent, et donc il devrait être satisfait. Nous avons de plus $i \notin \mathscr{N}_2$, car si c'était le cas, alors il y aurait un $j \in \mathscr{N}_1$ tel que $Obj^+(j) \subset Obj^+(i)$, et donc $Obj^+(j) \subseteq \mathscr{O} \setminus All(\overrightarrow{\pi})$, ce qui est impossible pour les mêmes raisons que ci-dessus. En conséquence, $\overrightarrow{\pi}$ est aussi Pareto-efficace, d'après le lemme 11. ▲

Après s'être penchés sur les deux restrictions naturelles sur le langage propositionnel utilisé pour l'expression des préférences dichotomiques, nous introduisons un résultat plus général. Ce résultat s'appuie sur le fait que le résultat de difficulté de la proposition 4.2 est très clairement lié à la NP-complétude du problème [SAT]. Que se passe-t-il si l'on se restreint, pour l'expression des préférences, à une certaine classe $\mathcal{C}$ telle que [SAT]($\mathcal{C}$) soit polynomial ? Dans le cas général où l'on ne fait aucune autre hypothèse sur $\mathcal{C}$, on ne peut rien affirmer de plus sur la complexité du problème [EEF EXISTENCE] que pour le problème général. En revanche, si nous ajoutons en plus le fait que $\mathcal{C}$ est clos pour la conjonction, la complexité du problème tombe dans NP :

Proposition 4.10 *Soit $\mathcal{C}$ une classe de formules propositionnelles close pour la conjonction telle que [SAT]($\mathcal{C}$) est dans* P*. Alors le problème [EEF EXISTENCE] pour des agents ayant des préférences dichotomiques exprimées uniquement avec des formules de la classe $\mathcal{C}$ est dans* NP.

Démonstration L'appartenance à NP vient du fait qu'après avoir deviné une allocation $\overrightarrow{\pi}$ de manière non déterministe, vérifier qu'elle est Pareto-efficace et sans envie peut être fait en temps polynomial. En effet, étant donnée une allocation, on peut vérifier qu'elle est sans envie en temps $O(nm)$ (où m est la longueur de la formule la plus grande), juste en vérifiant, pour chaque agent non satisfait, s'il aurait été satisfait avec la part d'un autre agent. Étant donné l'ensemble $Sat(\overrightarrow{\pi})$ des agents satisfaits par $\overrightarrow{\pi}$, vérifier la Pareto-efficacité de $\overrightarrow{\pi}$ revient à vérifier pour tout $i \in \mathscr{N} \setminus Sat(\overrightarrow{\pi})$ si $\bigwedge_{j \in Sat\overrightarrow{\pi}} \varphi_j \wedge \varphi_i$ est insatisfiable. Cela peut être fait à l'aide d'un nombre linéaire d'appels à un oracle [SAT]($\mathcal{C}$), puisque toutes les préférences sont dans $\mathcal{C}$, et que cette classe est close pour la conjonction. Cela prouve que le problème [EEF EXISTENCE] avec les préférences des agents dans $\mathcal{C}$ est

[1] Ou bien on aurait prouvé que P = NP, ce qui est peu réaliste.

dans NP. ▲

Un corollaire de cette proposition est que pour toute classe $\mathcal{C}$ de formules propositionnelles close pour la conjonction telle que [SAT]($\mathcal{C}$) est polynomial et qui contient les cubes, le problème [EEF EXISTENCE] est NP-complet. Cela s'applique par exemple à la classe des formules en des formules 2-CNF ou à la classe des clauses de Horn.

4.1.1.3 Critères d'efficacité alternatifs

La raison principale à la complexité élevée du problème [EEF EXISTENCE] est qu'il est difficile de vérifier qu'une allocation est Pareto-efficace. En conséquence, la complexité de ce problème peut décroître si nous abandonnons la Pareto-efficacité et que nous choisissons un autre critère pour l'efficacité. Nous allons nous intéresser à deux critères alternatifs de l'efficacité : la complétude de l'allocation, et le nombre maximal d'agents satisfaits.

Tout d'abord, nous nous intéressons à la complétude comme critère alternatif de la Pareto-efficacité. En d'autres termes, on demande seulement aux allocations d'être *complètes*. Sans surprise, la complexité du problème tombe dans NP.

Proposition 4.11 *Le problème d'existence d'une allocation complète et sans-envie pour des agents avec des préférences dichotomiques est* NP*-complet. Il reste* NP*-complet même si l'on fixe le nombre d'agents à 2, et que ces agents ont des préférences identiques.*

Démonstration Puisque l'on peut vérifier en temps polynomial qu'une allocation est complète, de même pour l'absence d'envie, l'appartenance à NP est directe.

Nous allons montrer la NP-complétude pour un problème à deux agents ayant des préférences identiques, par réduction depuis le problème [SAT]. Soit φ une formule propositionnelle. Nous créons l'instance du problème de partage qui suit : les objets correspondent aux symboles propositionnels de φ, et nous ajoutons un objet supplémentaire y ; les deux agents ont les mêmes préférences, représentées par la formule $\varphi \vee \mathbf{y}$. Il est immédiat de constater que tout partage complet satisfait au moins l'un des agents (celui qui se voit attribuer y). Si φ est satisfiable, alors il est possible de satisfaire aussi l'autre agent avec une part qui correspond à un modèle de φ : donc il existe un partage complet et sans envie. Réciproquement, supposons qu'il existe un partage complet et sans envie. L'un des deux agents est forcément satisfait grâce à φ, ce qui prouve qu'il existe un modèle de cette formule.

Nous avons donc prouvé la NP-complétude dans le cas de deux agents avec des préférences identiques. ▲

Nous pouvons remarquer que nous ne parlons pas dans la proposition du cas où les préférences sont monotones. Nous supposons que le problème d'existence d'un partage complet et sans envie pour deux agents ayant des préférences dichotomiques, identiques et monotones reste NP-complet (bien entendu il est dans NP), mais nous n'avons à ce jour pas la preuve de cette affirmation.

Le second critère alternatif d'efficacité auquel nous pouvons penser est le critère de maximalité pour la cardinalité (contrairement à la Pareto-efficacité qui est un critère de maximalité pour l'inclusion). Autrement dit, on recherche les allocations qui satisfont un nombre maximal d'agents.

Proposition 4.12 *Le problème d'existence d'une allocation sans envie parmi celles qui satisfont un* nombre maximal *d'agents avec des préférences dichotomiques et monotones est* Θ_2^{P}*-complet.*

Démonstration Le problème d'existence d'une allocation sans envie qui satisfait au moins k agents est dans NP. En conséquence, le nombre maximal d'agents pouvant être satisfaits simultanément peut être calculé par dichotomie en utilisant $\log n$ oracles NP. Il suffit ensuite, après avoir calculé ce nombre maximal d'agents, de deviner une allocation et de vérifier qu'elle est sans envie et satisfait le nombre d'agents calculé auparavant, ce qui ajoute un oracle NP supplémentaire. D'où l'appartenance à Θ_2^{p}.

La preuve de la Θ_2^{p}-complétude est obtenue par réduction depuis le problème suivant[2] :

Problème 5: $[\text{MAX-INDEX-SAT}]_{odd}$[Wagner, 1990]

INSTANCE :	Une suite de formules propositionnelles $(\chi_1, \ldots, \chi_n)$ telle que (χ_i est insatisfiable) $\Rightarrow$ (χ_{i+1} est insatisfiable.)
QUESTION :	L'index maximum i tel que χ_i est satisfiable est-il un nombre impair ?

Remarquons tout d'abord que la complexité de ce dernier problème reste la même sous les hypothèses suivantes :

- ▷ n est pair (s'il n'est pas pair, il suffit d'ajouter la formule $\perp$ à la fin de la liste) ;
- ▷ les ensembles de variables propositionnelles de la formule χ_i sont deux à deux disjointes (si deux formules χ_i et χ_{i+1} partagent des variables, il suffit de transformer chaque variable $\mathbf{x}$ de χ_i en une copie $\mathbf{x}'$ sans que cela ne change la satisfiabilité de χ_i, mais désormais les ensembles de variables propositionnelles de χ_i et χ_{i+1} sont disjoints).

Soit $(\chi_1, \ldots, \chi_n)$ une instance du problème $[\text{MAX-INDEX-SAT}]_{odd}$ avec les deux hypothèses additionnelles, et soit Var_i l'ensemble des variables propositionnelles apparaissant dans χ_i. Nous transformons cette instance en l'instance $\mathcal{P}(\chi_1, \ldots, \chi_n)$ définie comme suit :

Agents : $2n$ agents : $\mathscr{N} = \mathscr{N}_{1,2} \cup \mathscr{N}_{3,4} \cup \cdots \cup \mathscr{N}_{n-1,n}$, où le groupe $\mathscr{N}_{2i-1,2i}$ contient les quatre agents $\{4i-3, 4i-2, 4i-1, 4i\}$;

Objets : nous créons pour tout $\mathbf{x} \in Var_i$ (pour tout $i \in [\![1,n]\!]$) quatre objets o_v, $\overline{o_v}$, p_v, $\overline{p_v}$, et nous ajoutons n objets factices d_k ($k \in [\![1,2n]\!]$) ;

Préférences : pour chaque groupe $\mathscr{N}_{2i-1,2i}$ ($i \in [\![1,n/2]\!]$), les préférences des agents sont :

- ▷ $\varphi_{4i-3} = \varphi_{4i-2} = (\chi'_{2i-1} \wedge \mathbf{d_{2i-1}}) \vee (\chi'_{2i} \wedge \mathbf{d_{2i}})$,
- ▷ $\varphi_{4i-1} = \bigwedge_{\mathbf{x} \in V_{2i-1} \cup V_{2i}} \mathbf{o_x} \vee \overline{\mathbf{o_x}}$,
- ▷ $\varphi_{4i} = \bigwedge_{\mathbf{x} \in V_{2i-1} \cup V_{2i}} \mathbf{p_x} \vee \overline{\mathbf{p}}_\mathbf{x}$,

où χ'_k est la formule χ_k dans laquelle toute variable $\mathbf{x}$ a été remplacée par $\mathbf{o_x} \wedge \mathbf{p_x}$, et $\neg\mathbf{x}$ a été remplacée par $\overline{\mathbf{o}}_\mathbf{x} \wedge \overline{\mathbf{p}}_\mathbf{x}$.

La preuve de la proposition est fondée principalement sur le fait que le problème peut être découpé en $n/2$ sous-problèmes, chacun d'entre eux concernant les agents de $\mathscr{N}_{2i-1,2i}$:

Lemme 14 *Nous notons $\mathcal{P}^{|i}$ la restriction de $\mathcal{P}(\chi_1, \ldots, \chi_n)$ à l'ensemble des agents $\mathscr{N}_{2i-1,2i}$ et aux objets qu'ils désirent. Une allocation $\overrightarrow{\pi}$ sera dite* découpable *si $\forall i \neq j$, $\pi_{\mathscr{N}_{2i-1,2i}} \cap \pi_{\mathscr{N}_{2j-1,2j}} = \varnothing$. La restriction d'une allocation découpable $\overrightarrow{\pi}$ à $\pi_{\mathscr{N}_{2i-1,2i}}$ sera notée $\overrightarrow{\pi}^{|i}$.*

Il existe une allocation sans envie qui satisfait un nombre maximal d'agents de $\mathcal{P}(\chi_1, \ldots, \chi_n)$ si et seulement s'il existe une allocation découpable $\overrightarrow{\pi}$ telle que $\forall i \in [\![1,n/2]\!]$, $\overrightarrow{\pi}^{|i}$ est sans envie et satisfait un nombre maximal d'agents pour $\mathcal{P}^{|i}$.

[2]Problème qui est cité plusieurs fois dans la littérature, mais qui ne semble pas avoir de nom. Celui que nous lui donnons a été inventé.

Démonstration Nous nous restreignons tout d'abord aux allocations *régulières*. Par «régulières», nous entendons, comme dans le lemme 2, l'ensemble des allocations qui ne donnent un objet à un agent que s'il le désire. On peut se restreindre à ces allocations sans changer le problème, pour les mêmes raisons que dans le lemme 2 : l'existence d'une allocation sans envie qui satisfait un nombre maximal d'agents est équivalente à l'existence d'une allocation *régulière* sans envie qui satisfait un nombre maximal d'agents. Puisque les ensembles Var_i sont disjoints deux à deux, deux problèmes différents $\mathcal{P}^{|i}$ and $\mathcal{P}^{|j}$ n'ont aucun objet en commun, et donc toute allocation régulière est aussi découpable.

Soit $\overrightarrow{\pi}$ une allocation régulière. Supposons qu'il existe une allocation $\overrightarrow{\pi}'^{|i}$ pour le problème $\mathcal{P}^{|i}$, telle que $\overrightarrow{\pi}'^{|i}$ satisfait strictement plus d'agents que $\overrightarrow{\pi}^{|i}$. Alors l'allocation découpable construite à partir des sous-allocations $\overrightarrow{\pi}^{|j}$ pour $j \neq i$ et $\overrightarrow{\pi}'^{|i}$ est valide, régulière, et satisfait strictement plus d'agents que $\overrightarrow{\pi}$. Réciproquement, supposons qu'il existe une allocation régulière $\overrightarrow{\pi}'$ qui satisfait strictement plus d'agents que $\overrightarrow{\pi}$. Alors, il y a au moins un indice i tel que strictement plus d'agents de $\mathscr{N}_{2i-1,2i}$ sont satisfaits par $\overrightarrow{\pi}'^{|i}$ que par $\overrightarrow{\pi}^{|i}$. Cela prouve que toute allocation régulière $\overrightarrow{\pi}$ satisfait un nombre maximal d'agents si et seulement si pour tout i, $\overrightarrow{\pi}^{|i}$ satisfait un nombre maximal d'agents.

Supposons maintenant que $\overrightarrow{\pi}$ est sans envie. Alors de manière évidente tous les $\overrightarrow{\pi}^{|i}$ le sont. Réciproquement, supposons que tous les $\overrightarrow{\pi}^{|i}$ sont sans envie. Alors $\overrightarrow{\pi}$ est sans envie, car (1) aucun agent ne peut envier un autre agent du même groupe, car les $\overrightarrow{\pi}^{|i}$ sont sans envie, et (2) aucun agent d'un groupe i ne peut envier un agent d'un autre groupe j, car $\bigcup_{k \in \mathscr{N}_{2i-1,2i}} \pi_k \cap \bigcup_{k \in \mathscr{N}_{2j-1,2j}} \pi_k = \varnothing$. ▲

Pour toute interprétation v_k de Var_k, nous définissons les ensembles d'objets suivants :

- ▷ $\omega(v_k) = \{o_x \mid v_k \models x\} \cup \{\overline{o_x} \mid v_k \not\models x\}$;
- ▷ $\rho(v_k) = \{p_x \mid v_k \models x\} \cup \{\overline{p_x} \mid v_k \not\models x\}$;
- ▷ $\overline{\omega}(v_k) = \{o_x \mid v_k \not\models x\} \cup \{\overline{o_x} \mid v_k \models x\}$;
- ▷ $\overline{\rho}(v_k) = \{p_x \mid v_k \not\models x\} \cup \{\overline{p_x} \mid v_k \models x\}$.

De plus, étant données deux interprétations v_{2i-1} et v_{2i} de Var_{2i-1} et Var_{2i} respectivement, nous noterons $\overrightarrow{\pi}^{v_{2i-1},v_{2i}}$ l'allocation de $\mathcal{P}^{|i}$ définie comme suit :

- ▷ $\pi_{4i-3}^{v_{2i-1},v_{2i}} = \omega(v_{2i-1}) \cup \rho(v_{2i-1}) \cup \{d_{2i-1}\}$;
- ▷ $\pi_{4i-2}^{v_{2i-1},v_{2i}} = \omega(v_{2i}) \cup \rho(v_{2i}) \cup \{d_{2i}\}$;
- ▷ $\pi_{4i-1}^{v_{2i-1},v_{2i}} = \overline{\omega}(v_{2i-1}) \cup \overline{\omega}(v_{2i})$;
- ▷ $\pi_{4i}^{v_{2i-1},v_{2i}} = \overline{\rho}(v_{2i-1}) \cup \overline{\rho}(v_{2i})$.

Lemme 15 *Soient v_{2i-1} et v_{2i} deux interprétations respectives de Var_{2i-1} et Var_{2i}.*

- ▷ *$\overrightarrow{\pi}^{v_{2i-1},v_{2i}}$ satisfait les deux agents $4i-1$ et $4i$;*
- ▷ *$\overrightarrow{\pi}^{v_{2i-1},v_{2i}}$ satisfait $4i-3$ si et seulement si $v_{2i-1} \models \chi_{2i-1}$, et $\overrightarrow{\pi}^{v_{2i-1},v_{2i}}$ satisfait $4i-2$ si et seulement si $v_{2i} \models \chi_{2i}$;*

Démonstration Soient v_{2i-1} et v_{2i} deux interprétations respectives de Var_{2i-1} et Var_{2i}.

- ▷ Par définition, $\overline{\omega}(v_k)$ contient o_x ou $\overline{o_x}$ pour toute variable $x \in Var_k$, donc $\pi_{4i-1}^{v_{2i-1},v_{2i}}$ contient o_x ou $\overline{o_x}$ pour toute variable $v \in Var_{2i-1} \cup Var_{2i}$. En conséquence, l'agent $4i-1$ est satisfait par $\pi_{4i-1}^{v_{2i-1},v_{2i}}$. Le même raisonnement est valable pour l'agent $4i$.
- ▷ Par définition, χ_{2i-1} est satisfaite par v_{2i-1} si et seulement si χ'_{2i-1} est satisfaite par l'interprétation définie en instanciant à vrai tous les $\mathbf{o_x}$ et $\mathbf{p_x}$ (resp. tous les $\overline{\mathbf{o_x}}$ et $\overline{\mathbf{p_x}}$) tels que $v_{2i-1} \models \mathbf{x}$ (resp. $v_{2i-1} \not\models \mathbf{x}$). Donc si $v_{2i-1} \models \chi_{2i-1}$, $\pi_{4i-3}^{v_{2i-1},v_{2i}}$ satisfait χ'_{2i-1}. Puisque cette part satisfait aussi

$\mathbf{d_{2i-1}}$, $4i-3$ est donc satisfait par $\overrightarrow{\pi}^{v_{2i-1},v_{2i}}$. Réciproquement, si $4i-3$ est satisfait par $\overrightarrow{\pi}^{v_{2i-1},v_{2i}}$, alors clairement χ'_{2i-1} doit être satisfaite par $\pi_{4i-3}^{v_{2i-1},v_{2i}}$ (car $4i-3$ ne reçoit pas d_{2i}), ce qui prouve que χ_{2i-1} est satisfaite par v_{2i-1}. Le même raisonnement peut être appliqué pour χ_{2i} et l'agent $4i-2$. ▲

Lemme 16 *Considérons le problème restreint $\mathcal{P}^{|i}$.*

- ▷ *Si χ_{2i-1} et χ_{2i} sont toutes deux insatisfiables, alors pour toutes interprétations v_{2i-1} et v_{2i} de Var_{2i-1} et Var_{2i} respectivement, $\overrightarrow{\pi}^{v_{2i-1},v_{2i}}$ est sans envie et satisfait un nombre maximal d'agents.*
- ▷ *Si seule χ_{2i-1} est satisfiable, alors si M_{2i-1} est un modèle de χ_{2i-1}, $\overrightarrow{\pi}^{M_{2i-1},v_{2i}}$ satisfait un nombre maximal d'agents. De plus, il n'existe dans ce cas aucune allocation sans envie qui satisfait un nombre maximal d'agents.*
- ▷ *Si les deux formules χ_{2i-1} et χ_{2i} sont satisfiables, alors si M_{2i-1} et M_{2i} sont des modèles respectifs de χ_{2i-1} et χ_{2i}, $\overrightarrow{\pi}^{M_{2i-1},M_{2i}}$ satisfait un nombre maximal d'agents et est sans envie.*

Démonstration Supposons que ni χ_{2i-1} ni χ_{2i} ne sont satisfiables. Alors toute allocation $\overrightarrow{\pi}^{|i}$ qui satisfait $4i-3$ (respectivement $4i-2$) doit être telle qu'il existe au moins un $\mathbf{x} \in Var_{2i-1} \cup Var_{2i}$ tel que $\{o_x, \overline{o_x}, p_x, \overline{p_x}\} \subset \pi_{4i-3}^{|i}$ (respectivement $\pi_{4i-2}^{|i}$), car sinon on pourrait déduire un modèle de χ_{2i-1} ou χ_{2i} à partir de $\pi_{4i-3}^{|i}$ (respectivement $\pi_{4i-2}^{|i}$). En conséquence, aucun des agents $4i$ et $4i-1$ ne peut être satisfait dans ce cas : le nombre maximal d'agents qu'il est possible de satisfaire est 2. Puisque toute allocation de la forme $\overrightarrow{\pi}^{v_{2i-1},v_{2i}}$ satisfait les deux agents $4i-1$ et $4i$, une telle allocation satisfait un nombre maximal d'agents dans ce cas. Cette allocation est aussi de manière évidente sans envie, puisque ni d_{2i} ni d_{2i-1} ne sont dans les parts des agents $4i-1$ et $4i$, et donc les deux autres agents ne peuvent les envier.

Supposons que seule χ_{2i-1} est satisfiable. Alors toute allocation satisfaisant les deux agents $4i-3$ et $4i-2$ doit satisfaire χ'_{2i-1} pour l'un de ces deux agents, et χ'_{2i} pour l'autre (à cause de d_{2i} et d_{2i-1}). Puisque χ_{2i} n'est pas satisfiable, dans ce cas ni $4i-1$ ni $4i$ ne peuvent être satisfaits par $\overrightarrow{\pi}^{|i}$, pour les mêmes raisons que ci-dessus. Nous pouvons en déduire qu'il n'est pas possible de satisfaire les 4 agents en même temps. Il n'est pas possible non plus de satisfaire 3 agents avec à la fois $4i-3$ et $4i-2$ satisfaits. Maintenant considérons l'allocation $\overrightarrow{\pi}^{M_{2i-1},v_{2i}}$, M_{2i-1} étant un modèle de χ_{2i-1}. D'après le lemme 15, $\overrightarrow{\pi}^{M_{2i-1},v_{2i}}$ satisfait 3 agents : $4i-3$, $4i-1$ et $4i$. Cette allocation n'est pas sans envie, mais aucune allocation satisfaisant autant d'agents ne peut l'être dans ce cas (car soit $4i-3$ soit $4i-2$ reste insatisfait dans une telle allocation et donc envie son partenaire).

Enfin, supposons que les deux formules χ_{2i-1} et χ_{2i} sont satisfiables, et soient M_{2i-1} et M_{2i} leurs modèles. Alors d'après le lemme 15, $\pi^{M_{2i-1},M_{2i}}$ satisfait les 4 agents, satisfaisant ainsi un nombre maximal d'agents et étant de manière évidente sans envie. ▲

Nous pouvons à présent conclure la preuve. D'après le lemme 14, il existe une allocation sans envie parmi celles qui satisfont un nombre maximal d'agents pour $\mathcal{P}(\chi_1, \ldots, \chi_n)$ si et seulement s'il existe une allocation découpable $\overrightarrow{\pi}$ telle que $\forall i \in [\![1, n/2]\!]$, $\overrightarrow{\pi}^{|i}$ est sans envie et satisfait un nombre maximal d'agents pour $\mathcal{P}^{|i}$. D'après le lemme 15, il existe une allocation sans envie $\pi^{|i}$ qui satisfait un nombre maximal d'agents pour $\mathcal{P}^{|i}$ si et seulement si soit aucune des deux formules χ_{2i-1} et χ_{2i} n'est satisfiable, soit toutes les deux le sont. Maintenant supposons que l'indice maximum j tel que χ_j est satisfiable est un nombre impair (disons $2i - 1$). Dans ce cas, il n'y a aucune allocation sans envie qui satisfait un nombre maximal d'agents pour $\mathcal{P}^{|i}$ puisque χ_{2i-1} est satisfiable mais que χ_{2i} ne l'est pas. Réciproquement, supposons que l'indice j maximum tel que χ_j est satisfiable est un nombre pair (disons $2i$). Dans ce cas, il existe une allocation sans envie parmi celles qui satisfont un nombre maximal d'agents pour tout $\mathcal{P}^{|k}$, puisque pour tout $\mathcal{P}^{|k}$ soit les deux formules χ_{2k-1} et χ_{2k} sont satisfiables (si $k \leq i$), soit aucune d'entre elles ne l'est (si $k > i$).

Nous avons donc mis en évidence une réduction polynomiale du problème $[\mathsf{MAX\text{-}INDEX\text{-}SAT}]_{odd}$ vers le problème d'existence d'une allocation sans envie qui satisfait un nombre maximal d'agents, ce qui prouve la Θ_2^{p}-complétude. ▲

4.1.2 Préférences non-dichotomiques

4.1.2.1 Préférences logiques générales

Nous allons maintenant considérer le cas où les préférences ne sont plus dichotomiques. Encore une fois, comme nous nous sommes attachés à le démontrer au chapitre 3, nous avons besoin d'un langage de représentation compacte de préférences. Comme nous l'avons vu, de nombreux langages existent. Nous allons nous restreindre aux langages dérivés de la logique propositionnelle, ou plus précisément aux langages définis comme suit :

Définition 4.1 (Langage de représentation compacte sous forme logique) *Soit $\mathcal{R}_{\mathscr{O}}$ un langage de représentation compacte de préférences fondé sur $\mathscr{O}$. $\mathcal{R}_{\mathscr{O}}$ est un* langage compact sous forme logique *si et seulement si :*

(a) il est capable d'exprimer toute structure de préférence dichotomique de manière aussi compacte que le langage $\mathcal{R}_{dicho}$, c'est-à-dire que toute formule de $\mathcal{L}_{dicho}$ peut être transformée en temps polynomial en une formule de $\mathcal{L}_{\mathscr{O}}$;

(b) la comparaison de deux ensembles d'objets peut être effectuée en temps polynomial.

Ces deux conditions ne sont pas très restrictives en pratique, et sont vérifiées par de nombreux langages dédiés à la représentation compacte de préférences, tels que ceux que l'on a introduits dans le chapitre 3. De manière intéressante, la proposition 4.2 peut être étendue à n'importe quel langage de représentation de ce type :

Corollaire 2 *Le problème [EEF EXISTENCE] avec des agents ayant des préférences monotones exprimées de manière compacte sous forme logique est Σ_2^{p}-complet.*

Démonstration Le problème [EEF EXISTENCE] peut être résolu grâce à l'algorithme suivant :

1. deviner une allocation $\overrightarrow{\pi}$ de manière non-déterministe :
2. vérifier que cette allocation est sans envie ;
3. vérifier que cette allocation est Pareto-efficace.

D'après la condition (b), le deuxième pas de l'algorithme peut être effectué en temps polynomial, puisqu'il ne requiert qu'un nombre quadratique d'oracles polynômiaux. Selon la condition (b) à nouveau, le problème de vérification de la Pareto-efficacité d'une allocation est dans co-NP. Ainsi, l'algorithme non-déterministe précédent utilise un nombre polynomial d'oracles NP et s'exécute en temps polynomial. D'où l'appartenance à Σ_2^{p}.

La Σ_2^{p}-difficulté est une conséquence directe de la proposition 4.2 et de la condition (a). ▲

4.1.2.2 Préférences numériques sous forme logique

Pour ce dernier résultat, les préférences n'ont pas à être numériques, puisque la Pareto-efficacité et l'absence d'envie sont des notions purement ordinales. Maintenant, si les préférences sont numériques, ce qui implique la possibilité de les comparer et de les agréger, nous pouvons nous intéresser à un critère d'efficacité fondé sur la maximisation d'une fonction d'utilité collective à la place de la Pareto-efficacité. Nous allons nous concentrer sur les deux fonctions d'utilité collective les plus courantes, que nous avons introduites aux chapitre 1 : les fonctions d'utilité égalitariste et utilitariste classique, autrement dit la fonction min et la fonction somme.

Puisque nous ne sommes plus dans le cadre ordinal (ou dichotomique), nous nous devons de définir précisément ce que l'on entend par représentation compacte de préférences numériques. Nous allons bien entendu choisir le langage à base de formules pondérées, que nous avons introduit dans le chapitre 3, et qui est à la base du langage de représentation compacte du problème de partage qui a été introduit dans la définition 3.38. Rappelons brièvement la définition de ce langage.

Les préférences des agents sont représentées par un ensemble de formules logiques pondérées $\Delta_i = \{(\psi_{i,1}, w_{i,1}), \ldots, (\psi_{i,m_i}, w_{i,m_i})\}$. Nous supposerons ici que les poids sont dans $\mathbb{Z}$. Étant donné l'ensemble de formules précédent, l'utilité de l'agent i correspondant est définie comme suit :

$$u_i : \begin{array}{lcl} \wp(\mathscr{O}) & \to & \mathbb{Z} \\ \pi & \mapsto & \sum_{k=1}^{m} w_m \times \sigma(\Delta, \pi) \end{array} \qquad \sigma(\delta, \pi) = \begin{cases} w(\delta) & \text{si } \delta \text{ est satisfaite par } \pi, \\ \perp & \text{sinon.} \end{cases}$$

Comme nous l'avons déjà fait remarquer, les préférences exprimées dans ce langage sont monotones si et seulement si toutes les formules sont positives, de même que les poids.

Les résultats de complexité que nous allons introduire dans cette section sont centrés sur le langage de représentation sous forme logique pondérée, mais restent valable pour tout langage de représentation numérique compact «raisonnable» qui étend la logique pondérée. Plus précisément, ces résultats s'étendent à tout langage de représentation compacte numérique sous forme logique, cette notion étant définie comme suit :

Définition 4.2 (Langage de représentation compacte numérique sous forme logique) *Soit $\mathcal{R}_{\mathscr{O}}$ un langage de représentation compacte de préférences ordinales fondé sur $\mathscr{O}$. $\mathcal{R}_{\mathscr{O}}$ est un* langage compact numérique sous forme logique *si et seulement si :*

(a) il est capable d'exprimer toute structure de préférence dichotomique de manière aussi compacte que le langage $\mathcal{R}_{weighted}$, c'est-à-dire que toute formule de $\mathcal{L}_{weighted}$ peut être transformée en temps polynomial en une formule de $\mathcal{L}_{\mathscr{O}}$;

(b) la comparaison de deux ensembles d'objets peut être effectuée en temps polynomial.

Bien entendu, puisqu'un langage compact numérique sous forme logique est aussi un langage compact sous forme logique, le résultat de complexité du corollaire 2 reste valable. Cependant il apparaît que cette complexité décroît lorsque la Pareto-efficacité est remplacée par une notion plus faible : la maximisation de l'une des fonctions d'utilité collective égalitariste ou utilitariste classique.

Proposition 4.13 *Étant donnée une collection de fonctions d'utilité sur $\wp(\mathscr{O})$, spécifiées dans un langage compact numérique sous forme logique,*

- ▷ *le problème d'existence d'une allocation sans envie parmi celles qui maximisent la fonction d'utilité collective utilitariste classique est Δ_2^{p}-complet, même si le nombre d'agents est fixé à 2 et même si les agents ont des préférences identiques ;*
- ▷ *le problème d'existence d'une allocation sans envie parmi celles qui maximisent la fonction d'utilité collective égalitariste est Δ_2^{p}, même si le nombre d'agents est fixé à 2.*

Démonstration Pour ces deux résultats, l'appartenance à Δ_2^{P} est facile à démontrer, en considérant le fait que la valeur maximum de l'utilité collective peut être calculée par dichotomie sur l'ensemble de toutes les valeurs possibles de l'utilité collective. Puisqu'il y en a un nombre exponentiel, nous avons besoin d'un nombre polynomial d'oracles NP pour cela. Après cette opération, il suffit de deviner une allocation et de vérifier qu'elle est sans envie et qu'elle maximise l'utilité collective, ce qui ajoute simplement un autre oracle NP.

La difficulté est obtenue dans les deux cas utilitariste classique et égalitariste par une simple réduction vers une instance du problème de partage équitable avec préférences exprimées à l'aide de la logique pondérée, depuis le problème suivant :

Problème 6: [MAX-SAT-ASG]$_{even}$ [Wagner, 1987]

INSTANCE :	Une formule propositionnelle χ en forme normale conjonctive, sur un ensemble de variables propositionnelles $V = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$, et une fonction poids w sur les interprétations $v : \mathit{Var} \to \{0,1\}$, définie par $w(v) \stackrel{\text{def}}{=} \sum_i v(\mathbf{x_i}) \times 2^{i-1}$.
QUESTION :	Est-ce que $\max_{M \text{ modèle de } \chi} w(M)$ est un nombre pair (en d'autres termes la variable $\mathbf{x_1}$ est-elle falsifiée dans le modèle de poids maximal) ?

Nous allons supposer que la formule χ possède au moins un modèle M tel que $M \not\models \mathbf{x_1}$. Cela ne change rien à la complexité, car si $\mathbf{x_1}$ est vérifié dans tous les modèles de χ, la réponse au problème [MAX-SAT-ASG]$_{even}$ est clairement négative. En conséquence, toute instance (χ, V) sans aucune hypothèse sur χ peut être résolue en vérifiant d'abord si $\neg\mathbf{x_1} \wedge \chi$ est insatisfiable (c'est un problème co-NP-complet), et ensuite, si ce n'est pas le cas, en résolvant le problème [UNSAT-OR-MAX-SAT-ASG]$_{even}$ sur une instance qui possède au moins un modèle falsifiant $\mathbf{x_1}$.

Utilitarisme classique : À partir d'une instance (χ, V) du problème [MAX-SAT-ASG]$_{even}$, nous créons l'instance $\mathcal{P}(\chi, V)$ comme suit :

Agents : 2 agents ;

Objets : pour chaque littéral $\mathbf{x_i}$ de χ, nous créons deux objets o_i et o'_i, excepté pour $\mathbf{x_1}$, pour lequel nous n'introduisons qu'un seul objet o_1, et enfin nous ajoutons deux objets y et y' ;

Préférences : les agents 1 et 2 ont les préférences identiques suivantes : $((\psi \wedge \mathbf{y}) \vee (\psi' \wedge \mathbf{y}'), 2^{n+1}), (\mathbf{o_1} \wedge \mathbf{y}, 1), \ldots, (\mathbf{o_n} \wedge \mathbf{y}, 2^{n-1}), (\mathbf{o'_2} \wedge \mathbf{y}', 2), \ldots, (\mathbf{o'_n} \wedge \mathbf{y}', 2^{n-1})$, avec ψ la formule construite à partir de χ dans laquelle on a remplacé chaque symbole $\mathbf{x_i}$ par $\mathbf{o_i}$, et ψ' la formule construite à partir de χ dans laquelle on a remplacé chaque symbole $\mathbf{x_i}$ par $\mathbf{o'_i}$ (excepté $\mathbf{x_1}$ remplacé par $\mathbf{o_1}$).

Soit (M_1, M_2) un couple de modèles de χ (avec éventuellement $M_1 = M_2$) tel que $M_2 \not\models \mathbf{x_1}$. Nous définissons alors l'allocation $\overrightarrow{\pi}^{M_1,M_2}$ de la manière suivante :$\pi_1^{M_1,M_2} = \{y\} \cup \{o_i \mid M_1 \models \mathbf{x_i}\}$ et $\pi_2^{M_1,M_2} = \{y'\} \cup \{o'_i \mid M_2 \models \mathbf{x_i}\}$.

La preuve est fondée sur le lemme suivant :

Lemme 17 *Il existe une allocation sans envie parmi celles qui maximisent la fonction d'utilité collective utilitariste classique pour $\mathcal{P}(\chi, V)$ si et seulement s'il existe deux modèles M_1 et M_2 de χ (avec éventuellement $M_1 = M_2$) avec $M_2 \not\models \mathbf{x_1}$, tels que $\overrightarrow{\pi}^{M_1,M_2}$ est sans envie et maximise la fonction d'utilité collective utilitariste classique.*

Démonstration Soit $\overrightarrow{\pi}$ une allocation qui maximise l'utilité collective utilitariste classique. Soit M un modèle de χ qui falsifie $\mathbf{x_1}$ (notre hypothèse est qu'il en existe au moins un). Alors $F(\pi_1^{M,M}) \models \psi \wedge y$ et $F(\pi_2^{M,M}) \models \psi' \wedge y'$, ce qui prouve que l'utilité individuelle des deux agents est d'au moins 2^{n+1}. Ainsi il existe au moins une allocation dont l'utilité sociale utilitariste classique est au moins égale à 2^{n+2}. En conséquence, toute allocation $\overrightarrow{\pi}$ maximisant la fonction d'utilité collective utilitariste doit être telle que $F(\pi_1) \vDash y \wedge \psi$ et $F(\pi_2) \vDash y' \wedge \psi'$ ou vice-versa. De plus, soit $o_1 \notin \pi_1$, soit $o_1 \notin \pi_2$. Supposons que $o_1 \in \pi_2$: échanger les parts des agents conduit à une allocation $\overrightarrow{\pi}'$ qui est complètement équivalente du point de vue de l'utilité collective et de l'absence d'envie, à cause du fait que les préférences sont identiques. Nous pouvons donc supposer sans perte de généralité que $\overrightarrow{\pi}$ est tel que $o_1 \notin \pi_2$.

Puisque $\pi_1 \vDash \psi$, il existe un modèle M_1 de χ tel que $\pi_1 = \{y\} \cup \{o_i \mid M_1 \vDash \mathbf{x_i}\} \cup \mathscr{S}_1$, où $\mathscr{S}_1 \subseteq \{o_1, o'_2, \ldots, o'_n\}$. De manière similaire, il existe un modèle M_2 tel que $M_2 \nvDash \mathbf{x_1}$, et $\pi_2 = \{y'\} \cup \{o'_i \mid i > 1,\ M_2 \vDash \mathbf{x_i}\} \cup \mathscr{S}_2$, où $\mathscr{S}_2 \subseteq \{o_1, \ldots, o_n\}$. Considérons maintenant l'allocation $\overrightarrow{\pi}^{M_1,M_2}$, qui est bien définie, puisque $M_2 \nvDash \mathbf{x_1}$. On a $u_1(\overrightarrow{\pi}) = u_1(\overrightarrow{\pi}^{M_1,M_2})$, puisque les objets o'_i ne satisfont aucune formule de l'agent 1 sans y' (qui a été donné à l'agent 2), et $u_2(\pi) = u_2(\overrightarrow{\pi}^{M_1,M_2})$ pour les mêmes raisons. En d'autres termes, $\overrightarrow{\pi}^{M_1,M_2}$ donne la même utilité que $\overrightarrow{\pi}$ aux deux agents. $\overrightarrow{\pi}^{M_1,M_2}$ est donc sans envie et maximise l'utilité collective utilitariste classique. ▲

D'après le lemme 17, nous pouvons donc restreindre notre problème aux allocations de la forme $\overrightarrow{\pi}^{M_1,M_2}$. Nous avons, pour tous M_1 et M_2 définis comme précédemment, $u_1(\overrightarrow{\pi}^{M_1,M_2}) = 2^{n+1} + w(M_1)$ et $u_2(\overrightarrow{\pi}^{M_1,M_2}) = 2^{n+1} + w(M_2)$; donc $g^\star(\overrightarrow{\pi}^{M_1,M_2}) = 2^{n+2} + w(M_1) + w(M_2)$. Nous avons : $\text{argmax}_{\overrightarrow{\pi}^{M_1,M_2}} g^\star(\overrightarrow{\pi}^{M_1,M_2}) = \overrightarrow{\pi}^{\text{argmax}_{(M_1,M_2)}\{w(M_1)+w(M_2) \ \mid\ M_1 \nvDash \mathbf{x_1} \text{ ou } M_2 \nvDash \mathbf{x_1}\}}$. Étant donnée la symétrie du problème, nous pouvons supposer que seul M_2 doit satisfaire $M_2 \nvDash \mathbf{x_1}$, donc la précédente allocation devient : $\overrightarrow{\pi}^{M_{opt},\text{argmax}_{M_2}\{w(M_2) \ \mid\ M_2 \nvDash \mathbf{x_1}\}}$, où M_{opt} est le modèle de χ de poids maximal.

Supposons que $M_{opt} \nvDash \mathbf{x_1}$, alors l'allocation qui maximise l'utilité collective utilitariste est $\overrightarrow{\pi}^{M_{opt},M_{opt}}$, et elle est clairement sans envie, car les deux agents ont la même utilité. Maintenant supposons que $M_{opt} \vDash \mathbf{x_1}$. Dans ce cas, l'allocation qui maximise l'utilité collective utilitariste est $\overrightarrow{\pi}^{M_{opt},M_{opt'}}$, où $M_{opt'}$ est le modèle de χ de poids maximal qui falsifie $\mathbf{x_1}$. Nous avons $w(M_{opt'}) < w(M_{opt})$, et donc $u_1(\overrightarrow{\pi}^{M_{opt},M_{opt'}}) > u_2(\overrightarrow{\pi}^{M_{opt},M_{opt'}})$, donc l'allocation n'est pas sans envie.

Cette réduction est clairement polynomiale (rappelons que les poids 2^{n+1} peuvent être encodés sur un espace de taille linéaire). Cela prouve la proposition dans le cas utilitariste classique.

Égalitarisme : À partir d'une instance (χ, V) de $[\text{MAX-SAT-ASG}]_{even}$, nous créons l'instance $\mathcal{P}(\chi, V)$ de la manière suivante :

Agents : 2 agents ;

Objets : pour chaque littéral $\mathbf{x_i}$ de χ, nous créons deux objets o_i et o'_i, et nous ajoutons deux objets y et y' ;

Préférences : les préférences de l'agent 1 sont $(\mathbf{o_1}, 1), \ldots, (\mathbf{o_n}, 2^{n-1}), (\psi \wedge \mathbf{y}, 2^n)$, et les préférences de l'agent 2 sont $(\mathbf{y} \vee \mathbf{y'}, 2^{2n}), (\mathbf{o_1}, 1)$, avec ψ la formule χ dans laquelle chaque symbole $\mathbf{x_i}$ a été remplacé par $\mathbf{o_i}$.

Toute allocation qui maximise l'utilité collective égalitariste doit donner une utilité d'au moins 2^{2n} à l'agent 2. Dans ce cas, la valeur de l'utilité collective sera déterminée par l'utilité de l'agent 1, car son utilité ne peut pas être supérieure à 2^{2n}. En conséquence, maximiser l'utilité collective revient dans ce cas à maximiser l'utilité de l'agent 1, ou en d'autres termes à lui donner les objets qui correspondent au modèle de χ de poids maximum. Si $\mathbf{x_1}$ est instancié à vrai dans ce modèle, alors o_1 est donné à l'agent 1, et puisque y est aussi donné à cet agent, l'agent 2 pourrait avoir une utilité strictement supérieure avec la part de l'agent 1. En conséquence, cette allocation n'est pas sans envie. Si $\mathbf{x_1}$ est instancié à faux dans ce dernier modèle, alors o_1 n'est pas donné à l'agent 1 et peut donc être donné à l'agent 2, produisant ainsi une allocation sans envie. La réduction étant clairement polynomiale, nous avons donc montré la Δ_2^{p}-difficulté pour le cas égalitariste. ▲

Nous pouvons remarquer dans le cas utilitariste classique de la preuve précédente que le résultat de Δ_2^{p}-difficulté subsiste si nous remplaçons le critère de maximisation de l'utilité collective utilitariste classique par la Pareto-efficacité. Cela suggère donc que dans le cas d'un langage de représentation compacte logique numérique, le problème [EEF EXISTENCE] avec des préférences identiques (et un nombre d'agents fixé à 2) est beaucoup plus difficile que dans le cas où les préférences sont dichotomiques. Formellement, nous avons le résultat suivant :

Proposition 4.14 *Étant donnée une collection de n fonctions d'utilité identiques sur $\wp(\mathscr{O})$, spécifiées dans un langage compact numérique sous forme logique, le problème d'existence d'une allocation Pareto-efficace et sans envie est Δ_2^{p}-complet, même si $n = 2$ et que les préférences sont monotones.*

Démonstration Puisque les préférences sont identiques, toute allocation sans envie doit satisfaire les agents de manière égale. Ainsi, une allocation Pareto-efficace et sans envie, s'il y en a une, est une allocation qui donne une utilité de $\widehat{u}$ à tous les agents, et qui est maximale parmi l'ensemble des allocations qui satisfont tous les agents de manière égale. Cette valeur $\widehat{u}$ peut être calculée, comme dans la preuve précédente, à l'aide d'un nombre polynomial d'oracles NP. Ayant calculé cette valeur $\widehat{u}$, vérifier s'il existe une allocation Pareto-efficace et sans envie revient à vérifier s'il n'y a pas d'allocation qui donne une utilité d'au moins $\widehat{u}$ à tous les agents, et au moins $\widehat{u}+1$ à un agent. Ce dernier problème est dans co-NP, et donc on ne rajoute qu'un seul appel à un oracle NP.

Pour la preuve de Δ_2^{p}-difficulté, on peut remarquer que la même réduction que celle utilisée dans le cas utilitariste classique de la preuve de la proposition 4.13 fonctionne dans ce cas, car une allocation de cette instance particulière est Pareto-efficace et sans envie si et seulement si elle est sans envie et maximise le bien-être collectif utilitariste. ▲

4.1.2.3 Préférences numériques additives

Nous allons nous pencher sur un dernier cas : celui des préférences numériques *additives*. Ces préférences, que nous avons introduites dans la section 3.2.4.1 sous le nom de préférences *modulaires* (ou 1-additives), sont un cas dégénéré des préférences à base de logique pondérée, où toutes les formules sont atomiques (c'est-à-dire que toutes les formules sont des littéraux positifs). Comme nous l'avons vu, les préférences d'un agent i s'expriment dans ce cas comme un ensemble Δ_i de paires (o_k, w_k) , où o_k est un objet et w_k est le poids (potentiellement nul) associé à cet objet. L'utilité de l'agent i est calculée par sommation des poids correspondant aux objets qui sont dans la part de l'agent i. Bien entendu, les préférences des agents sont monotones si et seulement si tous les nombres w_k sont positifs.

Ce langage n'est bien sûr pas une extension du langage des préférences dichotomiques. En conséquence, le résultat de difficulté précédent de s'étend pas aux préférences additives. Bien entendu, puisque nous sommes toujours capables de comparer deux alternatives en temps polynomial, l'appartenance à Σ_2^{p} est garantie.

Se pose donc la question de la complexité exacte du problème [EEF EXISTENCE] avec des préférences numériques. Notre intuition est que, malgré la simplicité apparente de ce langage de représentation de préférences, ce problème est aussi difficile que le problème [EEF EXISTENCE] avec des préférences dichotomiques :

Conjecture 1 *Le problème [EEF EXISTENCE] avec des préférences additives numériques est Σ_2^{p}-complet, même si les préférences sont monotones.*

Tout ce que nous savons à propos de ce problème est qu'il est NP-difficile et qu'il appartient à Σ_2^{p}, mais sa complexité exacte reste inconnue à ce jour. Cependant, les choses sont beaucoup plus faciles si nous remplaçons le critère de Pareto-efficacité par la complétude de l'allocation. Ce cas a déjà été étudié dans la littérature [Lipton *et al.*, 2004], et nous avons le résultat suivant :

Proposition 4.15 [Lipton *et al.*, 2004] *Le problème d'existence d'une allocation complète et sans envie pour des agents ayant des préférences numériques additives est* NP*-complet, même si les préférences sont monotones.*

D'autres restrictions du problème [EEF EXISTENCE] avec des préférences additives valent la peine d'être étudiées. Tout d'abord, nous allons nous intéresser comme dans le cas dichotomique la restriction aux préférences additives identiques :

Proposition 4.16 *Le problème [EEF EXISTENCE] avec n agents ayant des préférences numériques additives identiques est* NP*-complet, pour tout $n \geq 2$ fixé. Ce résultat reste valable si nous nous restreignons aux préférences monotones.*

Démonstration L'appartenance à NP est facile à démontrer. Puisque toutes les préférences sont identiques (nous notons $(f(o_1), \ldots, f(o_p))$ le vecteur de poids associé au vecteur des objets $(o_1, \ldots, o_p)$), une allocation est Pareto-efficace si et seulement si elle attribue chaque objet o_j tel que $f(o_j) > 0$ à un agent, et laisse de côté tous les objets o_j tels que $f(o_j) \leq 0$. De plus, une allocation est sans envie si et seulement si elle donne la même utilité à tous les agents. Ces deux propriétés peuvent être vérifiées en temps polynomial, d'où l'appartenance à NP.

Nous allons montrer la difficulté du problème par une réduction depuis le problème [PARTITION] :

Problème 7: [PARTITION]

INSTANCE :	Un ensemble fini $\mathscr{S}$ et une taille $s(a) \in \mathbb{N}$ pour tout $a \in \mathscr{S}$.
QUESTION :	Existe-t-il un sous-ensemble $\mathscr{S}' \subseteq \mathscr{S}$ tel que $\sum_{a \in \mathscr{S}'} s(a) = \sum_{a \in \mathscr{S} \setminus \mathscr{S}'} s(a)$?

À partir d'une instance donnée $(\mathscr{S}, s)$ du problème [PARTITION], nous créons l'instance $\mathcal{P}(\mathscr{S}, s)$ du problème [EEF EXISTENCE] comme suit :

Agents :	2 agents ;
Objets :	nous associons à chaque $a \in S$, un objet o_a ;
Préférences :	Les préférences des deux agents sont identiques et définies par la fonction taille sur les éléments de l'ensemble initial : $f(o_a) = s(a)$.

Il existe une allocation Pareto-efficace et sans envie pour l'instance $\mathcal{P}(\mathscr{S}, s)$ si et seulement s'il existe une allocation $\overrightarrow{\pi}$ telle que $\sum_{o \in \pi_1} f(o) = \sum_{o \in \pi_2} f(o)$, c'est-à-dire si et seulement si $(\mathscr{S}, s)$ est une instance positive du problème [PARTITION]. La réduction est clairement polynomiale, ce qui prouve la proposition. ▲

Intéressons-nous à un cas encore plus dégénéré de préférences numériques : le cas où les préférences sont additives, mais où toutes les utilités atomiques $f_i(o_j)$ (désignant le poids attribué à l'objet j par l'agent i) sont égales à 0 ou 1. En d'autres termes, soit un agent désire un objet, soit il ne le désire pas. Chaque agent essaie de maximiser le nombre d'objets désirés qu'il obtient.

Proposition 4.17 *Le problème [EEF EXISTENCE] avec des préférences additives 0–1 (c'est-à-dire* $\forall i, j,\ f_i(o_j) \in \{0, 1\}$*) est* NP*-complet.*

Démonstration La Pareto-efficacité est aisée à vérifier dans ce cas. Nous pouvons tout d'abord enlever les objets qui n'apparaissent nulle part dans les préférences sans changer le problème. Après cette opération, une allocation est Pareto-efficace si et seulement si chaque objet o_j est donné à un agent i tel que $f_i(o_j) = 1$, et ce pour les raisons suivantes. ($\Rightarrow$) Soit $\overrightarrow{\pi}$ une allocation Pareto-efficace, et supposons qu'il existe un o_j qui n'est donné à aucun agent, ou bien qui est donné à un agent i tel que $f_i(o_j) = 0$. Soit k un agent tel que $f_k(o_j) = 1$ (il y en a forcément un puisque nous avons laissé de côté les objets non désirés lors de la dernière opération). Alors le fait de donner l'objet o_j à l'agent k augmente son utilité sans changer l'utilité des autres agents. Donc $\overrightarrow{\pi}$ est Pareto-dominé. ($\Leftarrow$) Soit $\overrightarrow{\pi}$ une allocation telle que tout objet o_j est donné à un agent i tel que $f_i(o_j) = 1$, et supposons que $\overrightarrow{\pi}$ est Pareto-dominée par une allocation $\overrightarrow{\pi}'$ donnée. Alors $\sum_{i \in \mathscr{N}} f_i(\pi'_i) = \sum_{i \in \mathscr{N}} \sum_{o_j \in \pi'_i} f_i(o_j) > \sum_{i \in \mathscr{N}} f_i(\pi_i) = \sum_{i \in \mathscr{N}} \sum_{x_j \in \pi'_i} f_i(x_j) = p$. Donc il existe au moins un $f_i(o_j)$ tel que $f_i(o_j) > 1$, ce qui est impossible en vertu de notre restriction aux préférences 0–1.

Cela nous donne donc un moyen simple de vérifier la Pareto-efficacité, juste en calculant la somme des utilités et en déterminant si elle est égale au nombre p d'objets qui sont désirés par au moins un agent. Comme à l'accoutumée, l'absence d'envie peut être vérifiée en temps polynomial ; donc le problème [EEF EXISTENCE] avec des préférences additives 0–1 est dans NP.

La complétude peut être démontrée par une réduction polynomiale depuis le problème [EXACT COVER BY 3-SETS], dont nous rappelons la définition :

Problème 4 (rappel): [EXACT COVER BY 3-SETS] [Karp, 1972]

INSTANCE :	Un ensemble $\mathscr{S}$ de taille $3q$, et une collection $C = (\mathscr{C}_1, \ldots, \mathscr{C}_{\vert C\vert})$ de sous-ensembles à 3 éléments de $\mathscr{S}$
QUESTION :	C contient-il une couverture exacte pour $\mathscr{S}$, c'est-à-dire une sous-collection $C' \subseteq C$ telle que chaque élément de $\mathscr{S}$ apparaît dans exactement un membre de C' ?

Étant donnée une instance $(\mathscr{S}, C = (\mathscr{C}_1, \ldots, \mathscr{C}_{|C|}))$ du problème [EXACT COVER BY 3-SETS], nous créons l'instance $\mathcal{P}(C, \mathscr{S})$ du problème [EEF EXISTENCE] définie comme suit (nous supposerons que les éléments de $\mathscr{S}$ sont notés a_i, avec $i \in [\![1, |S|]\!]$) :

Agents : un ensemble de $3|C|$ agents regroupés par triplets $\{3i-2, 3i-1, 3i\}$;

Objets : un ensemble de $|\mathscr{S}| + 3|C|$ objets $\mathscr{O} = \mathscr{M} \cup \mathscr{D}$ ($\mathscr{M}$ pour «*main*», principaux, et $\mathscr{D}$ pour «*dummy*», factices), avec $\mathscr{M} = \{m_1, \ldots, m_{|\mathscr{S}|}\}$, et $\mathscr{D} = \bigcup_{i \in [\![1,|C|]\!], j \in \{1,2,3\}} \{d_{i,j}\}$;

Préférences : les agents de $\{3i-2, 3i-1, 3i\}$ désirent tous le même ensemble d'objets $\bigcup_{a_k \in \mathscr{C}_i} \{m_k\} \cup \{d_{i,1}, d_{i,2}, d_{i,3}\}$ (les trois objets correspondant à $\mathscr{C}_i$ plus les trois objets factices $d_{i,j}$).

S'il existe une couverture exacte C' pour l'instance $(C, \mathscr{S})$, alors nous allons considérer l'allocation suivante : chaque agent du triplet $\{3i-2, 3i-1, 3i\}$ obtient respectivement $d_{i,1}$, $d_{i,2}$, et $d_{i,3}$, et si $\mathscr{C}_i \in C'$, chacun de ces trois agents obtient l'un des trois objets m_k correspondant aux éléments de l'ensemble $\mathscr{C}_i$. Cette allocation est admissible et Pareto-efficace (car tous les objets sont alloués). Elle est aussi sans envie, pour les raisons suivantes :

- ▷ Les agents du même triplet ne peuvent pas s'envier les uns les autres, car ils sont tous également satisfaits.

- ▷ Un agent k_1 ne peut envier un agent k_2 faisant partie d'un autre triplet que lui, puisque les seuls objets que k_1 peut envier dans la part de k_2 sont les m_i. k_2 obtenant au plus un seul m_i, et k_1 ayant une utilité au moins égale à 1, k_1 ne peut donc pas envier k_2.

La suite de la preuve est fondée sur le résultat suivant : si une allocation $\overrightarrow{\pi}$ est Pareto-efficace et sans envie pour $\mathcal{P}(C, \mathscr{S})$ alors on doit avoir $\pi_{3i-2} \cup \pi_{3i-1} \cup \pi_{3i} = \bigcup_{a_k \in \mathscr{C}_i} \{m_k\} \cup \{d_{i,1}, d_{i,2}, d_{i,3}\}$ ou $\pi_{3i} \cup \pi_{3i-1} \cup \pi_{3i} = \{d_{i,1}, d_{i,2}, d_{i,3}\}$. Ce résultat est facile à démontrer. Puisque les agents du triplet $\{3i-2, 3i-1, 3i\}$ sont les seuls à désirer les objets $d_{i,k}$, ces trois objets doivent être donnés à ces trois agents, pour que cette allocation soit efficace. Puisque ces trois agents ont les mêmes préférences, l'allocation doit les satisfaire de manière égale pour qu'elle soit sans envie. Ainsi, le nombre d'objets alloués aux trois agents doit être divisible par 3, ce qui donne seulement deux nombres possibles, 3 et 6, et donc seulement deux allocations possibles.

Supposons qu'il y ait une allocation Pareto-efficace et sans envie $\overrightarrow{\pi}$ pour $\mathcal{P}(C, \mathscr{S})$. Considérons la sous-collection $C' = \{\mathscr{C}_1, \ldots, \mathscr{C}_{|C'|}\}$ constituée des triplets $\mathscr{C}_i$ de la collection C tels que $\pi_{3i-2} \cup \pi_{3i-1} \cup \pi_{3i} = \bigcup_{a_k \in \mathscr{C}_i} \{m_k\} \cup \{d_{i,1}, d_{i,2}, d_{i,3}\}$. Nous avons alors les résultats suivants.

- ▷ Les $\mathscr{C}_i$ sont disjoints deux à deux. Supposons qu'il y ait un couple (i, j) tel que $i \neq j$ et tel qu'il existe un élément a_k appartenant à la fois à $\mathscr{C}_i$ et $\mathscr{C}_j$. Alors m_k est attribué à deux agents différents : un membre du triplet $\{3i-2, 3i-1, 3i\}$, et un membre du triplet $\{3j+1, 3j+2, 3j+3\}$, ce qui est impossible..
- ▷ $\bigcup_{i \in [\![1,|C'|]\!]} \mathscr{C}_i = \mathscr{S}$. Soit a_k un élément de $\mathscr{S}$. Puisque $\overrightarrow{\pi}$ est Pareto-efficace, m_k doit être alloué à un agent qui le désire (disons que cet agent appartient au triplet $\{3j+1, 3j+2, 3j+3\}$), à moins que personne ne le désire, ce qui arrive lorsque $\bigcup_{i \in [\![1,|C|]\!]} \mathscr{C}_i \neq \mathscr{S}$. Alors, d'après le résultat précédent, tous les objets de $\bigcup_{a_l \in \mathscr{C}_j} \{m_l\}$ doivent être alloués à ce triplet. En conséquence, $\mathscr{C}_j \in C'$. Puisque $a_k \in \mathscr{C}_j$, a_k appartient à au moins un élément de la collection C'.

En conséquence, C' est une couverture exacte de $\mathscr{S}$, ce qui prouve au final la proposition.
▲

La proposition 4.17 et la conjecture 1 mettent en évidence l'existence d'un grand fossé de complexité (au moins si la conjecture est vraie) entre le problème pour lequel nous autorisons des poids quelconques, et le problème pour lequel nous imposons des poids 0 ou 1. La question naturelle que cela suscite est de savoir si cette chute de complexité est spécifique aux préférences 0–1 ou si elle subsiste dès que l'on fixe une borne supérieure pour les poids.

Conjecture 2 *La complexité du problème* [EEF EXISTENCE] *avec des préférences additives 0–1–…–k pour $k \geq 2$ fixé est aussi élevée que celle du problème général avec des préférences additives non bornées.*

La détermination de la complexité précise de ce problème reste un problème ouvert, mais comme cela est suggéré dans la conjecture, notre intuition est que ce problème est aussi difficile que le problème [EEF EXISTENCE] avec des préférences non bornées.

Un autre problème naturel est soulevé par la proposition 4.17 : quelle est la complexité du problème [EEF EXISTENCE] avec des préférences 0–1 *stratifiées* ? Par «préférences 0–1 stratifiées», nous entendons le fait que les préférences sont données par un ensemble de couples (o_k, p), où o_k est un objet et p est un niveau de priorité. La comparaison de deux ensembles d'objets s'effectue en comparant de manière lexicographique les vecteurs dont les composantes représentent, pour chaque indice i, le nombre d'objets de priorité i dans la part d'un agent. Notons que ce problème n'est pas une instance du problème [EEF EXISTENCE] avec des préférences additives, ni une instance du problème [EEF EXISTENCE] avec des préférences numériques sous forme logique. S'il est relativement évident de vérifier que ce problème reste dans Σ_2^{p}, en revanche, sa complexité précise reste inconnue à ce jour.

Enfin, nous allons nous pencher sur le cas où le nombre d'objets en jeu est plus petit que le nombre d'agents. On pourrait penser que dans ce cas, le problème devient trivial. Il l'est en effet pour des préférences monotones, mais il ne l'est apparemment plus si les préférences sont non monotones.

Proposition 4.18 *Soit $\mathcal{P}$ un problème d'allocation avec n agents ayant des préférences numériques additives et monotones. Supposons que tous les agents désirent au moins un objet, et soit p le nombre d'objets désirés par au moins un agent.*

- ▷ *Si $p < n$, alors il n'existe aucune allocation Pareto-efficace et sans envie.*
- ▷ *Si $p = n$, le problème de l'existence d'une allocation efficace et sans envie pour des agents ayant des préférences monotones et additives est dans* P.

Démonstration ▷ $p < n$ **:** Chaque objet étant désiré par au moins un agent, toute allocation Pareto-efficace est complète. Si le nombre d'agents p est strictement inférieur au nombre d'agents n, alors au moins un agent i est insatisfait. En conséquence, il y a un agent j qui obtient un objet désiré par i, ce qui crée de l'envie de la part de i. En conséquence, aucune allocation Pareto-efficace ne peut être sans envie.

▷ $p = n$ **:** Puisqu'il y a autant d'agents que d'objets, chaque agent doit recevoir un objet parmi ceux qu'il évalue le plus cher (c'est-à-dire un objet tel que $f_i(x)$ est maximal) pour qu'une allocation soit efficace et sans envie. En effet, si un agent i reçoit un objet qui n'est pas parmi ses préférés, cela veut dire (puisque toute allocation Pareto-efficace est complète) qu'un autre agent l'a reçu, suscitant ainsi l'envie de i. En conséquence, vérifier l'existence d'une allocation Pareto-efficace et sans envie se ramène dans ce cas à vérifier s'il est possible de donner à chaque agent un objet parmi ceux qu'il évalue le plus cher. Cela se ramène donc à vérifier s'il existe un couplage parfait dans le graphe constitué d'un nœud par agent d'un côté et d'un nœud par objet de l'autre, un arc connectant un nœud-agent i à un nœud-objet o si et seulement si l'objet o est parmi les objets qui ont la valuation la plus haute dans les préférences de i. Un tel couplage parfait peut être calculé en temps polynomial, d'où le résultat. ▲

Si le résultat précédent paraît relativement trivial, il est en revanche intéressant de constater qu'il ne tient plus du tout pour des préférences non monotones. Pis encore, dans ce cas, le problème est aussi complexe que le problème [EEF EXISTENCE] général avec des préférences additives, ce qui est pour le moins surprenant.

Proposition 4.19 *Le problème [EEF EXISTENCE] avec des préférences additives numériques et tel que le nombre d'objets est plus petit que le nombre d'agents a la même complexité que le problème [EEF EXISTENCE] avec des préférences numériques additives et aucune hypothèse sur le nombre d'objets.*

Démonstration Soit $(\mathscr{N}, \mathscr{O}, (\ldots, f_i(o_j), \ldots))$ une instance du problème [EEF EXISTENCE] avec des préférences numériques additives, n agents et p objets ($p > n$). Nous créons l'instance $\mathcal{P}(\mathscr{N}, \mathscr{O}, (\ldots, f_i(o_j), \ldots))$ du problème [EEF EXISTENCE] définie comme suit :

Agents :	$p + 3$ agents (le nombre d'agents n'est pas important, pourvu qu'il soit plus élevé que le nombre d'objets et que le nombre d'agents initial ;
Objets :	les p objets d'origine o_i plus deux objets factices o_1 et o_2 ;

Préférences : les préférences des n premiers agents sont les mêmes que dans l'instance initiale $(\mathscr{N}, \mathscr{O}, (\dots, f_i(o_j), \dots))$; le préférences du $(n+1)^{\text{ème}}$ agent sont $f_{n+1}(d_1) = f_{n+1}(d_2) = 1$ et $f_{n+1}(o_j) = 0$ pour les autres objets o_j, et les préférences des agents restants sont $f_j(d_1) = 1$, $f_j(d_2) = -2$ et $f_j(\{o_j\}) = 0$ pour les objets restants.

S'il existe une allocation efficace et sans envie $\overrightarrow{\pi}$ pour $(\mathscr{N}, \mathscr{O}, (\dots, f_i(o_j), \dots))$ alors on peut vérifier simplement que l'allocation qui donne les mêmes objets aux n premiers agents de $\mathcal{P}(\mathscr{N}, \mathscr{O}, (\dots, f_i(o_j), \dots))$, $\{d_1, d_2\}$ au $(n+1)^{\text{ème}}$ agent, et rien du tout aux agents restants est efficace et sans envie. Réciproquement, toute allocation Pareto-efficace et sans envie pour $\mathcal{P}(\mathscr{N}, \mathscr{O}, (\dots, f_i(o_j), \dots))$ conduit à une allocation Pareto-efficace et sans envie pour $(\mathscr{N}, \mathscr{O}, (\dots, f_i(o_j), \dots))$ en restreignant cette allocation aux n premiers agents et à tous les objets sauf les objets factices. ▲

4.1.3 Conclusion

Nous avons donc identifié la complexité exacte du problème d'existence d'une allocation efficace et sans envie lorsque les préférences des agents sont représentées sous forme compacte dans différents contextes : pour différentes notions de l'efficacité (Pareto-efficacité, complétude, nombre maximal d'agents, maximisation d'une fonction d'utilité collective), pour différents types de préférences (dichotomiques ou non) et pour diverses restrictions. Les résultats de complexité obtenus sont résumés dans la figure 4.1 et dans le tableau 4.1.

La grande complexité de ce problème d'existence dans de nombreux cas étudiés ici amène à se poser la question de la pertinence du critère d'absence d'envie. Certains travaux envisagent la question sous l'angle d'une approximation numérique du critère, fondée sur diverses définitions de la notion d'envie [Lipton *et al.*, 2004; Chevaleyre *et al.*, 2007a]. Une autre approche, fondée sur la définition d'un critère d'envie à portée limitée et prenant en compte une information incomplète des agents (un agent ne peut envier que des agents qui lui sont proches ou qu'il connaît) semble particulièrement intéressante, et n'a pas encore été étudiée, à notre connaissance, dans la littérature.

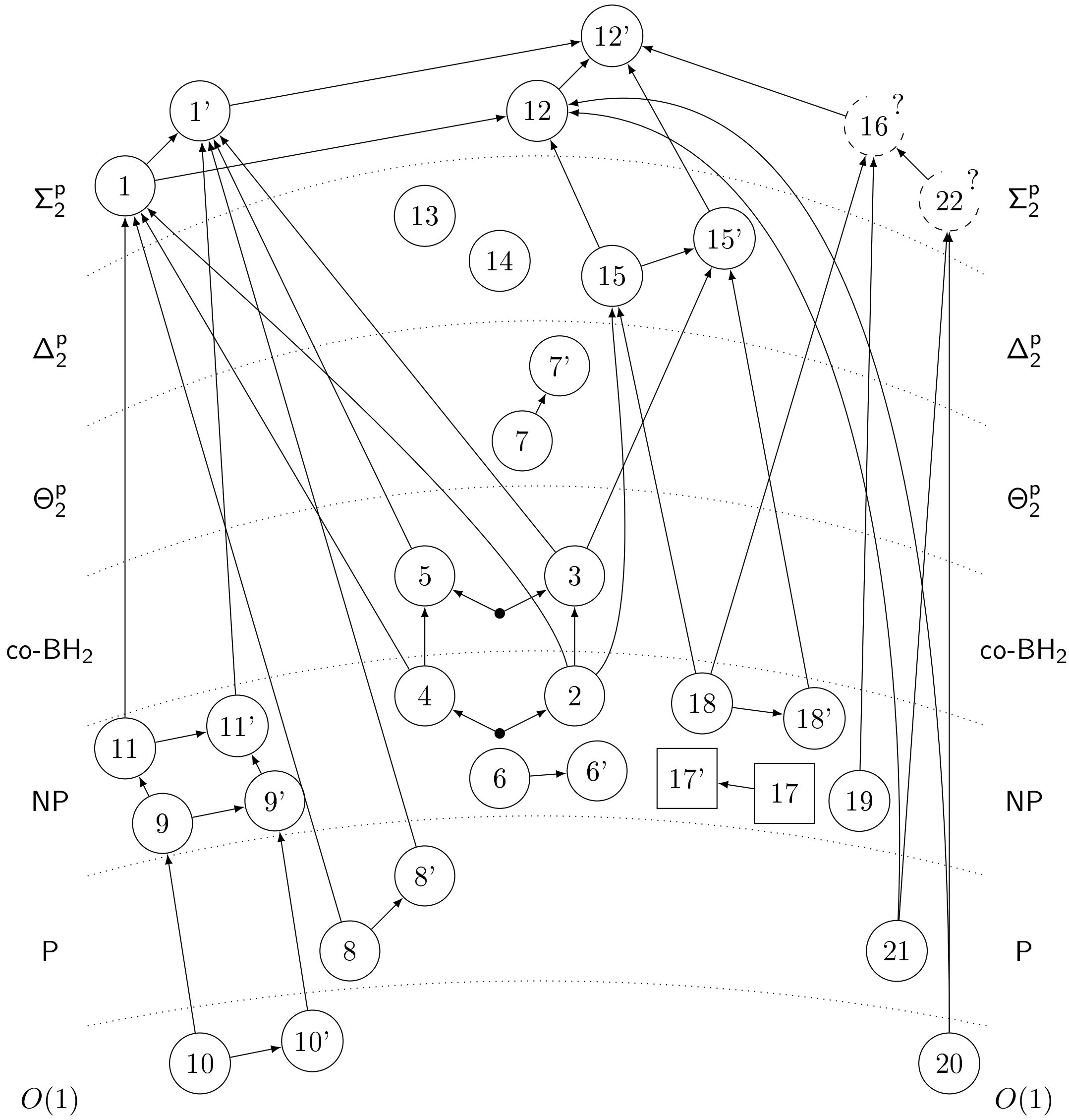
4.2 Maximisation de l'utilité collective

Nous allons maintenant nous intéresser à la complexité du deuxième problème introduit dans le chapitre 3 (définition 3.38), fondé sur une expression des préférences en logique pondérée, sur une expression logique des contraintes, et sur la définition d'une fonction d'utilité collective. Plus précisément, nous nous intéressons au problème de décision suivant :

Problème 8: [MAX-CUF]

INSTANCE : Une instance $(\mathscr{N}, \mathscr{O}, (\Delta_1, \dots, \Delta_n), \mathscr{C}, \langle \mathscr{V}_{ind}, \geq_{ind}, \oplus \rangle, \langle \mathscr{V}, \geq, \rangle, g)$ du problème de partage de biens indivisibles et un élément $K \in \mathscr{V}$.
QUESTION : Existe-t-il une allocation admissible $\overrightarrow{\pi}$ telle que $u_c(\overrightarrow{\pi}) \geq K$?

Par la suite, nous supposerons que toutes les utilités sont exprimées par des entiers, autrement dit que $\mathscr{V}_{ind} = \mathscr{V} = \mathbb{N}$. Bien entendu, cette restriction n'affecte pas vraiment la portée des résultats de complexité introduits ici, ni leur importance. Nous ferons cependant une exception à cette hypothèse afin de pouvoir représenter l'ordre social leximin : pour ce problème particulier, nous supposerons que $\mathscr{V}_{ind} = \mathbb{N}$, $\mathscr{V} = \mathbb{N}^n$, $\succeq = \succeq_{leximin}$, et $g = Id$. Autrement dit, le problème de maximisation de l'utilité collective se ramène dans ce cas-là au problème de recherche des partages admissibles dont



1 — Problème dont nous avons prouvé la complexité (la correspondance entre les nombres et les problèmes est spécifiée dans le tableau 4.1).

17 — Problème dont la complexité était déjà connue dans la littérature.

22 ? — Problème dont la complexité reste inconnue.

• — Intersection des problèmes correspondant aux arcs sortants.

$i \rightarrow j$ — Le problème i est inclus dans le problème j. Les arcs qui peuvent être obtenus par transitivité ont été omis.

Figure 4.1 — Les différents problèmes de décision concernant l'absence d'envie, et leurs classes de complexité et relations d'inclusion.

	Efficacité	**nombre d'agents**	**préférences**	**monotonie**	**complexité**
			Préférences dichotomiques		
1, 1'	Pareto-eff.	non fixé	—	oui (1) ou non (1')	Σ_2^p-c.
2	Pareto-eff.	non fixé ou fixé avec $n \geq 2$	identiques	oui	NP-c.
3	Pareto-eff.	non fixé ou fixé avec $n \geq 2$	identiques	non	co-BH_2-c.
4	Pareto-eff.	2 agents	—	oui	NP-c.
5	Pareto-eff.	2 agents	—	non	co-BH_2-c.
6, 6'	all. complète	non fixé ou fixé avec $n \geq 2$	identiques ou pas	oui (6) ou non (6')	NP-c.
7, 7'	nb max d'agents	non fixé	—	oui (7) ou non (7')	Θ_2^p-c.
8, 8'	Pareto-eff.	—	disjonctions	oui (8) ou non (8')	P
9, 9'	Pareto-eff.	—	conjonctions	oui (9) ou non (9')	NP-c.
10, 10'	Pareto-eff.	—	conjonctions avec condition 4.1	oui (10) ou non (10')	$O(1)$
11, 11'	Pareto-eff.	—	$\mathcal{C}$ tq [SAT] ($\mathcal{C}$)$\in$ P et clos pour $\wedge$	oui (11) ou non (11')	NP-c.
			Préférences non dichotomiques		
12, 12'	Pareto-eff.	non fixé	numériques	oui (12) ou non (12')	Σ_2^p-c.
13	utilitarisme cl.	non fixé ou fixé avec $n \geq 2$	numériques	non	Δ_2^p-c.
14	égalitarisme	non fixé ou fixé avec $n \geq 2$	numériques	non	Δ_2^p-c.
15, 15'	Pareto-eff.	non fixé ou fixé avec $n \geq 2$	numériques, identiques	oui (15) ou non (15')	Δ_2^p-c.
16	Pareto-eff.	non fixé	additives	non	Σ_2^p-c. ?
17, 17'	all. complète	non fixé	additives	oui (17) ou non (17')	NP-c.
18, 18'	Pareto-eff.	non fixé ou fixé avec $n \geq 2$	additives identiques	oui (18) ou non (18')	NP-c.
19	Pareto-eff.	—	additives 0–1	oui	NP-c.
20	Pareto-eff.	$>$ Nb d'objets	additives	oui	$O(1)$
21	Pareto-eff.	$=$ Nb d'objets	additives	oui	P
22	Pareto-eff.	$\geq$ Nb d'objets	additives	non	Σ_2^p-c. ?

Tableau 4.1 — L'ensemble des problèmes de partage dont la complexité a été étudiée dans cette section. Le résumé de leurs classes de complexité est représenté dans la figure 4.1.

les profils d'utilité individuelle sont non dominés pour l'ordre leximin. Pour simplifier les notations, nous noterons dans ce cas $g = $ leximin.

4.2.1 Complexité du problème général

Pour commencer, nous nous intéressons à la complexité du problème général [MAX-CUF], sans restriction sur les contraintes ou les fonctions d'agrégation individuelle et collective. Il est immédiat d'après la définition 3.38 de constater qu'il est toujours possible de vérifier en temps polynomial si une allocation est admissible ou non. Nous ferons de plus l'hypothèse (non précisée dans cette même définition) que l'utilité collective d'une allocation peut aussi être calculée en temps polynomial. Cette hypothèse est raisonnable dans la mesure ou elle est vérifiée pour tous les opérateurs d'agrégation individuelle introduits lors de la définition du langage de représentation au chapitre 3 et pour toutes les fonctions d'utilité collective introduites au chapitre 1.

Ainsi, sous les hypothèses introduites ici, le problème [MAX-CUF] est dans NP, quelles que soient les contraintes d'admissibilité ou les demandes formulées par les agents. Il s'avère que dans le cas général, où l'on ne fait aucune restriction sur les contraintes, ce problème est NP-complet :

Proposition 4.20 (Problème général [MAX-CUF]) *Le problème [MAX-CUF] est* NP*-complet dès lors que l'introduction de n'importe quelle contrainte est autorisée.*

> **Démonstration** La preuve de cette proposition se fait par simple réduction du problème [SAT]. À toute formule propositionnelle φ dont les symboles sont $Var(\varphi)$ nous associons le problème d'allocation défini par $\mathscr{N} = \{1\}$, $\mathscr{O} = Var(\varphi) \cup \{o'\}$, avec $o' \notin Var(\varphi)$, $\mathscr{C} = \{\varphi\}$, $\Delta_1 = \{(o', 1)\}$, et des fonctions d'agrégation $\oplus$ et g complètement quelconques (mais monotones). On peut immédiatement remarquer qu'il existe un partage admissible si et seulement si φ est satisfiable. Donc il existe un partage $\overrightarrow{\pi}$ admissible tel que $u_c(\overrightarrow{\pi}) \geq 0$ si et seulement si φ est satisfiable : nous avons réduit une instance quelconque du problème [SAT] en une instance du problème [MAX-CUF]. La réduction est clairement polynomiale, ce qui prouve la proposition. ▲

Nous avons donc démontré la difficulté du problème dans le cas général où l'on ne s'impose aucune restriction sur les contraintes, et ce pour n'importe quel couple d'opérateurs d'agrégation fixés. Il est en conséquence naturel de s'interroger sur les cas pour lesquels la complexité de ce problème de décision tombe dans P, moyennant l'introduction d'hypothèses restrictives sur les contraintes ou les demandes, et selon le choix des opérateurs d'agrégation $\oplus$ et g. Nous considérons ici seulement le cas des opérateurs d'agrégation les plus classiques, c'est-à-dire $\oplus = +$ ou max, et $g = +$, min, ou leximin. Comme nous allons le voir, la plupart des cas analysés se ramènent à des instances de problèmes bien connus en théorie de la complexité.

4.2.2 Pas de contrainte

Nous commençons par analyser un cas peu intéressant : celui d'un problème de partage sans contrainte :

Proposition 4.21 (Problème [MAX-CUF] non contraint) *Toute instance du problème [MAX-CUF] dans laquelle $\mathscr{C} = \varnothing$ peut être résolue en temps polynomial.*

> **Démonstration** Bien entendu, ce résultat utilise l'hypothèse de monotonie des fonctions $\oplus$ et g, et le fait que l'on se restreint à des demandes positives. Considérons une

instance du problème [MAX-CUF] dans laquelle l'ensemble des contraintes est vide et considérons l'allocation $\widehat{\overrightarrow{\pi}}$ qui donne tous les objets à chaque agent. Cette allocation est admissible (nous rappelons que la contrainte de préemption n'est pas présente puisque l'ensemble des contraintes est vide), et maximise l'utilité collective, en vertu des hypothèses de monotonie des opérateurs d'agrégation et de positivité des demandes. On peut calculer en temps polynomial $u_c(\widehat{\overrightarrow{\pi}})$ et donc déterminer dans le même temps s'il existe une allocation admissible dont l'utilité collective est supérieure ou égale à K. ▲

4.2.3 Contraintes de préemption uniquement

Nous allons maintenant considérer le cas, relativement classique et donc plus intéressant que le cas précédent, d'une instance du problème [MAX-CUF] ne contenant aucune autre contrainte que la contrainte de préemption. Comme nous allons le voir, ce problème, qui semble très basique, est toutefois aussi difficile que le problème général, et ce même si l'on se restreint au choix des fonctions d'agrégation classiques décrits ci-avant.

Proposition 4.22 (Contrainte de préemption uniquement) *Le problème [MAX-CUF] restreint aux instances sans autre contrainte que la contrainte de préemption est* NP*-complet, même dans les cas particuliers où les opérateurs d'agrégation sont les suivants :*
- ▷ $g = +$;
- ▷ $\oplus = +$ *ou* max, *et* $g = +$ *ou* min *ou* leximin.

Démonstration Bien entendu, l'appartenance à NP est immédiate. Il reste donc à prouver la difficulté pour les cas particuliers cités dans la proposition.
$g = +$ **(sans restriction sur $\oplus$) :** Nous allons utiliser une réduction depuis le problème [INDEPENDENT SET] :

Problème 9: [INDEPENDENT SET]	
INSTANCE :	Un graphe $\mathcal{G} = (V, E)$ et un entier K.
QUESTION :	Existe-t-il un sous-ensemble $\mathscr{S} \subseteq V$ de taille au moins K tel que pour tout couple (v, v') d'éléments de $\mathscr{S}$, $(v, v') \notin E$ (autrement dit, v et v' ne sont pas connectés dans $\mathcal{G}$) ?

Soit $((V, E), K)$ une instance du problème [INDEPENDENT SET], avec $V = \{v_1, \ldots, v_n\}$. Nous considérons l'instance $\mathcal{P}((V, E), K)$ du problème [MAX-CUF] définie de la manière suivante :
- ▷ $\mathscr{N} = [\![1, n]\!]$;
- ▷ $\mathscr{O} = \{o_{\{v_i,v_j\}} \mid (v_i, v_j) \in E\}$ (nous créons un objet pour chaque arête du graphe initial ; précisons aussi que $o_{\{v_i,v_j\}} = o_{\{v_j,v_i\}}$) ;
- ▷ pour tout agent i, $\Delta_i = (\varphi_i, 1)$, avec $\varphi_i = \bigwedge_{(v_i,v_j)\in E} o_{\{v_i,v_j\}}$ (la conjonction de tous les objets correspondant aux arcs incidents de v_i) ;
- ▷ $\mathscr{V} = \mathbb{N}$, $\oplus$ est quelconque (les agents n'ayant qu'une seule demande chacun, l'opérateur d'agrégation individuelle n'est pas important), et $g = +$;
- ▷ on cherche un partage dont l'utilité collective est supérieure à K.

Supposons qu'il existe un ensemble indépendant $\mathscr{S}$ de taille plus grande que K dans le graphe (V, E). Alors soit $\overrightarrow{\pi}(\mathscr{S})$ l'allocation qui attribue à tous les agents i tels que $v_i \in \mathscr{S}$ l'ensemble des objets de leur conjonction. Cette allocation est admissible : si elle ne l'était pas, cela signifierait que deux agents i et j satisfaits par $\overrightarrow{\pi}(\mathscr{S})$ ont un objet en commun, donc que les nœuds qui leur correspondent dans (V, E) ont une arête adjacente en commun, ou en d'autres termes qu'ils sont connectés. Cela est rendu impossible par le fait que $\mathscr{S}$

est un ensemble indépendant. En outre, $\overrightarrow{\pi}(\mathscr{S})$ satisfait autant d'agents que le nombre de nœuds dans $\mathscr{S}$, donc $u_c(\overrightarrow{\pi}(\mathscr{S})) \geq K$.

Réciproquement, supposons qu'il existe une allocation admissible $\overrightarrow{\pi}$ de $\mathcal{P}((V, E), K)$ dont l'utilité collective est supérieure ou égale à K. Il y a donc au moins K agents dont la part contient la conjonction des objets $o_{\{v_i,v_j\}}$ qu'ils désirent. Notons $\mathscr{S}(\overrightarrow{\pi})$ l'ensemble des nœuds v_i de V tels que i est satisfait par $\overrightarrow{\pi}$, et supposons qu'il existe deux nœuds v_i et v_j de $\mathscr{S}(\overrightarrow{\pi})$ qui sont connectés dans (V, E). Dans ce cas, l'objet $o_{\{v_i,v_j\}} = o_{\{v_j,v_i\}}$ apparaît dans la conjonction de l'agent v et dans celle de l'agent v', donc aussi dans leur part respective, ce qui est impossible en vertu de la contrainte de préemption. Donc $\mathscr{S}(\overrightarrow{\pi})$ est tel qu'il ne contient aucune paire de nœuds connectés, et est de taille supérieure ou égale à K, ce qui prouve au final la validité de la réduction.

$\oplus = \max$ **et** $g = \min$ **ou** leximin **:** Cette fois-ci, nous utilisons une réduction depuis le problème [SET PACKING] :

Problème 10: [SET PACKING]	
INSTANCE :	Une collection C d'ensembles finis, et un entier K.
QUESTION :	Existe-t-il une sous-collection $C' \subseteq C$ d'ensembles disjoints tels que $\lvert C'\rvert \geq K$?

Soit (C, K) une instance du problème [SET PACKING]. Nous considérons l'instance $\mathcal{P}(C, K)$ du problème [MAX-CUF] définie de la manière suivante :

- ▷ $\mathscr{N} = \{1, \ldots, K\}$;
- ▷ $\mathscr{O} = \bigcup_{\mathscr{S} \in C} \mathscr{S}$;
- ▷ pour tout agent i, $\Delta_i = \bigcup_{\mathscr{S} \in C}\{(\varphi_C, 1)\}$, avec $\varphi_C = \bigwedge_{o \in C} o$ (la conjonction de tous les objets du sous-ensemble C) ;
- ▷ $\mathscr{V} = \mathbb{N}$, $\oplus = \max$ et $g = \min$;
- ▷ on cherche un partage dont l'utilité collective est supérieure à 1.

Supposons que l'instance (C, K) soit une instance positive du problème [SET PACKING], et considérons un sous-ensemble $C' \subseteq C$ d'ensembles disjoints tel que $|C'| \geq K$. Nous définissons une allocation $\overrightarrow{\pi}(C')$ telle que $\forall i$, $\pi_i(C') \in C'$ (la part d'un agent i correspond à un élément particulier de C'), et $\forall j \neq i$, $\pi_i(C') \neq \pi_j(C')$ (les parts de deux agents différents correspondent à deux éléments différents de C'). Puisque $|C'| \geq K$, cette allocation est bien définie. Puisque les ensembles de C' sont disjoints, $\overrightarrow{\pi}(C')$ satisfait la contrainte de préemption. Enfin, puisque chaque agent voit l'une de ses demandes satisfaites, $u_c(\overrightarrow{\pi}(C')) = 1$.

Réciproquement, supposons qu'il existe une allocation $\overrightarrow{\pi}$ telle que $u_c(\overrightarrow{\pi}) \geq 1$. Alors l'utilité individuelle est au moins égale à 1 pour chaque agent, ce qui signifie que chaque agent i voit au moins une de ses formules pondérées $(\varphi_{C_i}, 1)$ satisfaite. Soit C' la collection des sous-ensembles C_i. Alors très clairement C_i est une sous-collection de C par définition des préférences des agents. De plus, les ensembles de C' sont mutuellement disjoints car $\overrightarrow{\pi}$ respecte la contrainte de préemption. En outre, C' contient K éléments. C'est donc une sous-collection de C d'ensembles mutuellement disjoints et de taille K, ce qui prouve la validité de la réduction utilisée.

Enfin, notons que dans le problème $\mathcal{P}(C, K)$, toutes les solutions maximin-optimales ont un profil d'utilité de $(1, \ldots, 1)$, donc elles sont toutes équivalentes au sens du leximin. En conséquence, l'ensemble des solutions maximin-optimales dans $\mathcal{P}(C, K)$ est égal à l'ensemble des solutions leximin-optimales, ce qui montre que la réduction précédente fonctionne aussi dans le cas leximin. ▲

Notons que ce résultat était déjà connu dans le cas particulier $(\oplus, g) = (+, g)$, puisqu'il s'agit d'un corollaire de la NP-difficulté du *Winner Determination Problem* dans les enchères combinatoires [Rothkopf *et al.*, 1998] ; ainsi, le problème [MAX-CUF] était déjà connu comme étant NP-complet dans ce cas. Cependant, les autres cas n'ont jamais été étudiés dans la littérature à notre connaissance.

Dans les cas précédents, une partie de la complexité semble provenir du fait que les agents ont des demandes complexes. On peut donc légitimement se demander ce qu'il advient de la complexité du problème [MAX-CUF] lorsque les demandes des agents sont additivement indépendantes, autrement dit, lorsque les formules pondérées sont atomiques.

Proposition 4.23 (Contrainte de préemption uniquement et demandes atomiques) *Le problème [MAX-CUF] restreint aux instances sans autre contrainte que la contrainte de préemption et pour lesquelles les préférences des agents sont atomiques est* NP*-complet, même dans le cas particulier où il n'y a que deux agents ayant des préférences identiques, avec des opérateurs d'agrégation* $(+, \min \mid \text{leximin})$.

Cependant, ce même problème est dans P *dans les cas suivants : (a)* $(\oplus, g) = (+, +)$ *; (b)* $(\oplus, g) = (\max, + \mid \min)$ *; (c)* $(\oplus, g) = (+, \min)$ *et les demandes ont toutes les mêmes poids.*

Démonstration Nous commençons par démontrer la NP-difficulté dans le cas $(+, \min \mid \text{leximin})$, ce qui prouvera bien-sûr la NP-complétude dans le cas général où l'on ne fait aucune hypothèse sur les opérateurs d'agrégation.

$\oplus = +$ **et** $g = \min$ **ou** leximin **:** La NP-difficulté du problème va être démontrée par réduction depuis le problème [PARTITION] :

Problème 7 (rappel): [PARTITION]

INSTANCE :	Un ensemble fini $\mathscr{S}$ et une taille $s(a) \in \mathbb{N}$ pour tout $a \in \mathscr{S}$.
QUESTION :	Existe-t-il un sous-ensemble $\mathscr{S}' \subseteq \mathscr{S}$ tel que $\sum_{a \in \mathscr{S}'} s(a) = \sum_{a \in \mathscr{S} \setminus \mathscr{S}'} s(a)$?

Soit $(\mathscr{S}, s)$ une instance du problème [PARTITION], avec $\mathscr{S} = \{a_1, \ldots, a_p\}$. Nous considérons l'instance $\mathcal{P}(\mathscr{S}, s)$ du problème [MAX-CUF] définie de la manière suivante :

▷ $\mathscr{N} = \{1, 2\}$;
▷ $\mathscr{O} = \{o_1, \ldots, o_p\}$ (nous créons un objet pour chaque élément de $\mathscr{S}$) ;
▷ $\Delta_1 = \Delta_2 = \{(o_1, s(a_1)), \ldots (o_p, s(a_p))\}$;
▷ $\mathscr{V} = \mathbb{N}$, $\oplus = +$ et $g = \min$;
▷ on cherche un partage dont l'utilité collective est supérieure à $K = \sum_{i=1}^{p} s(a_i)/2$.

En vertu de la contrainte de préemption, tout partage admissible complet correspond à une partition des objets $\mathscr{O}$ en deux sous-ensembles, et donc correspond à une partition des éléments de $\mathscr{S}$ en deux sous-ensembles (disjoints par définition). Puisque $\sum_{i=1}^{p} s(a_i) = 2K$, on a $u_1(\overrightarrow{\pi}) + u_2(\overrightarrow{\pi}) \leq 2K$ pour tout $\overrightarrow{\pi}$, et $u_1(\overrightarrow{\pi}) + u_2(\overrightarrow{\pi}) = 2K$ si et seulement si $\overrightarrow{\pi}$ est complet. En conséquence, il existe un partage $\overrightarrow{\pi}$ tel que $uc(\overrightarrow{\pi}) \geq K$ si et seulement si $u_1(\overrightarrow{\pi}) = u_2(\overrightarrow{\pi}) = K$, c'est-à-dire si et seulement s'il existe une partition des éléments de $\mathscr{S}$ en deux sous-ensembles $\mathscr{S}'$ et $\mathscr{S} \setminus \mathscr{S}'$ tels que $\sum_{a \in \mathscr{S}'} s(a) = \sum_{a \in \mathscr{S} \setminus \mathscr{S}'} s(a) = K$. Nous avons prouvé le résultat dans le cas particulier où il n'y a que deux agents ayant des préférences identiques.

Bien entendu, cette réduction fonctionne aussi dans le cas où $g = \text{leximin}$.

Nous allons maintenant nous intéresser à un certain nombre de cas pour lesquels le problème est polynomial.

$\oplus = +$ **et** $g = +$ **:** Ce cas se ramène de manière évidente à un cas d'enchères standard (non combinatoires), et la solution est évidente : attribuer chaque objet à un agent qui l'évalue le plus cher de tous les agents conduit à une allocation optimale. D'où l'appartenance de ce problème à P.

$\oplus = \max$ **et** $g = +$ **:** Puisque $\oplus = \max$, nous pouvons nous restreindre aux allocations admissibles n'attribuant qu'un seul objet à chaque agent : nous pouvons reconnaître aisément un problème de couplage de poids maximal :

Problème 11: [MAXIMAL MATCHING]

INSTANCE :	Un graphe bipartite pondéré $\mathcal{G}(V = N_1 \cup N_2, E)$, avec $w(v, v')$ désignant le poids d'une arête $\{v, v'\}$, et un entier K.
QUESTION :	Existe-t-il un couplage de poids supérieur ou égal à K, c'est-à-dire un sous-ensemble $E' \subseteq E$ tel que $\sum_{\{v,v'\} \in E'} w(v, v') \geq K$, et $\forall v \in V$, $\nexists (v', v'') \in V^2$ tel que $\{v, v'\} \in E'$ et $\{v, v''\} \in E'$?

Pour une instance du problème [MAX-CUF] avec opérateurs $\oplus = \max$ et $g = +$, nous créons le graphe bipartite formé d'un côté des agents et de l'autre des objets. Dans ce graphe, une arête relie un objet à un agent si l'objet en question figure dans les préférences de cet agent. Le poids d'une telle arête correspond au poids de l'objet dans les préférences de l'agent. Tout couplage dans un tel graphe correspond à une allocation admissible attribuant au plus un objet à chaque agent. Le poids d'un tel couplage correspond à l'utilité collective de l'allocation associée. Il y a donc une allocation d'utilité collective supérieure ou égale à un entier K si et seulement s'il y a un couplage de poids supérieur ou égal à K dans le graphe bipartite. Le problème [MAXIMAL MATCHING] étant dans P, c'est donc aussi le cas pour le problème [MAX-CUF] dans ce cas particulier.

$\oplus = \max$ et $g = \min$: Une réduction depuis le problème [MAXIMAL MATCHING] similaire au cas précédent fonctionne dans ce cas. La construction du graphe à partir d'une instance de [MAX-CUF] s'effectue de la même manière, sauf que le poids d'une arête ne correspond pas au poids de l'objet dans les demandes de l'agent, mais est égal à 1 si ce poids est supérieur ou égal à K et 0 sinon. On cherche cette fois-ci un couplage de poids au moins n, c'est-à-dire un couplage parfait. S'il existe, il est possible de trouver une allocation admissible qui donne à chaque agent un objet d'utilité au moins K, donc il existe un partage admissible d'utilité collective au moins K. Sinon, il n'est pas possible de trouver un tel partage, donc l'utilité collective maximale est 0.

Nous pouvons remarquer en revanche que cette réduction ne fonctionne pas pour le cas $g = \text{leximin}$. La complexité exacte de ce problème reste encore indéterminée.

$\oplus = +$ et $g = \min$, avec demandes de même poids : Ce cas ressemble beaucoup au premier cas dont nous avons démontré la NP-complétude. En revanche, le fait que les demandes soient toutes de même poids fait chuter la complexité et rend le problème polynomial. Voyons pourquoi. Tout d'abord, si les demandes sont toutes de poids w et que l'on cherche un partage d'utilité collective supérieure ou égale à K, nous pouvons transformer le problème en divisant les utilités par w : on se ramène ainsi à un problème (équivalent) de recherche d'un partage d'utilité collective supérieure ou égale à $K' = \lceil K/w \rceil$ dans une instance pour laquelle les poids de toutes les demandes sont égaux à 1. Nous pouvons réduire ce problème à un problème de flot dans le graphe $\mathcal{G} = (V, E)$ suivant :

- ▷ $V = \mathcal{N} \cup \mathcal{O} \cup \{s, t\}$;
- ▷ $E = \{(s, i, K') | i \in \mathcal{N}\} \cup \{(o, t, 1) | o \in \mathcal{O}\} \cup \{(i, o, 1) | i \in \mathcal{N}, o \in \mathcal{O}, \text{ et } (o, 1) \in \Delta_i\}$ (autrement dit, une arête de poids K' reliant la source s à tous les nœuds agents, une arête de poids 1 reliant tous les nœuds objets au puits, et une arête de poids 1 par couple (i, o) tel que o apparaît dans les demandes de l'agent i).

À chaque allocation admissible $\overrightarrow{\pi}$ telle que chaque agent ne reçoit pas plus de K' objets correspond un flot dans le graphe : pour chaque arête (s, i), la valeur du flot dans cette arête est le nombre d'objets obtenus par l'agent i (autrement dit, son utilité) ; pour chaque arête (i, o), la valeur du flot dans cette arête est 1 si $o \in \pi_i$, et 0 sinon ; pour chaque arête (o, t), la valeur du flot dans cette arête est 1 si o est donné à un agent, et 0 sinon. On peut vérifier facilement que ce flot est valide, et qu'il y a une bijection entre l'ensemble des flots valides dans le graphe $\mathcal{G}$ et l'ensemble des partages admissibles.

Supposons qu'il existe un partage admissible $\overrightarrow{\pi}$ d'utilité collective supérieure ou égale à K'. Alors il existe un partage admissible $\overrightarrow{\pi}'$ tel que chaque agent reçoit exactement K' objets (si un agent reçoit plus de K' objets dans $\overrightarrow{\pi}$, on peut lui en enlever jusqu'à arriver à K'). À l'allocation $\overrightarrow{\pi}'$ correspond un flot de $\mathcal{G}$, de valeur nK'. Réciproquement, s'il existe un flot de valeur nK' dans $\mathcal{G}$, alors il existe un partage admissible $\overrightarrow{\pi}$ tel que chaque agent reçoit exactement K' objets, et donc $u_c(\overrightarrow{\pi}) = K'$. Nous avons montré la validité de la réduction, ce qui prouve que le problème est dans P. ▲

4.2.4 Contraintes de volume uniquement

Comme nous l'avons précisé lors de l'introduction du modèle dans le chapitre 3, nous allons ici relâcher temporairement l'hypothèse d'expression des contraintes d'admissibilité en logique propositionnelle pure pour nous intéresser à la complexité des problèmes en présence de contraintes de volume. La notion de contrainte de volume a été introduite dans la section 1.1.2, lors de la définition de la notion de contrainte d'admissibilité.

Proposition 4.24 (Contraintes de volume uniquement) *Le problème [MAX-CUF] restreint aux instances sans autre contrainte que des contraintes de volume est* NP*-complet, même dans les cas particuliers où les demandes sont atomiques, et où les opérateurs d'agrégation* $(\oplus, g)$ *sont* $(+, (*))$, $((*), +)$ *ou* $(\max, \min \,|\, \text{leximin})$*, où* $(*)$ *désigne n'importe quel opérateur fixé.*

Démonstration **$\oplus = +$ ou $g = +$:** Pour ces cas, relativement identiques, on reconnaît aisément une instance du problème sac-à-dos :

Problème 12: [KNAPSACK]

INSTANCE :	Un ensemble fini $\mathscr{S}$ une fonction de valeur $u : \mathscr{S} \to \mathbb{N}$, une fonction de volume $v : \mathscr{S} \to \mathbb{N}$, une capacité maximale v_{max} et un entier K.
QUESTION :	Existe-t-il un sous-ensemble $\mathscr{S}' \subseteq \mathscr{S}$ tel que $\sum_{a \in \mathscr{S}'} v(a) \leq v_{max}$ et $\sum_{a \in \mathscr{S}'} u(a) \geq K$?

À partir d'un tel problème, on peut créer une instance du problème [MAX-CUF], avec un seul agent (donc l'opérateur g n'importe pas) ayant des demandes atomiques du type $(a, u(a))$ pour chaque élément $a \in \mathscr{S}$, l'opérateur d'agrégation individuel $\oplus = +$, et une contrainte de volume pesant sur tous les objets, le volume de chaque objet a étant égal à $v(a)$, et le volume maximal étant égal à v_{max}. Pour ce qui est du cas $((*), +)$, il suffit de remplacer l'agent unique par autant d'agents que d'objets, chaque agent désirant un seul objet.

$\oplus = \max$ et $g = \min$: On montre la NP-complétude dans ce cas par réduction depuis le problème [HITTING SET] :

Problème 13: [HITTING SET]

INSTANCE :	Une collection C de sous-ensembles $\mathscr{S}_1, \ldots, \mathscr{S}_n$ d'un ensemble $\mathscr{S}$ et un entier $K \leq |\mathscr{S}|$.
QUESTION :	Existe-t-il un sous-ensemble $\mathscr{S}' \subseteq \mathscr{S}$, avec $|\mathscr{S}'| \leq K$, tel que $\mathscr{S}' \cap \mathscr{S}_i \neq \varnothing$ pour tout $\mathscr{S}_i \in C$?

À partir d'une instance $(C = \{\mathscr{S}_1, \ldots, \mathscr{S}_n\}, K)$ de ce problème, nous créons l'instance de [MAX-CUF] suivante :

- ▷ $\mathscr{N} = \{1, \ldots, n\}$;
- ▷ $\mathscr{O} = \mathscr{S}$;

- ▷ une contrainte de volume telle que $v(o) = 1$ pour tous les objets de $\mathscr{O}$, et le volume maximal est K ;
- ▷ pour tout i, $\Delta_i = \{(o, 1) | o \in \mathscr{S}_i\}$;
- ▷ $\mathscr{V} = \mathbb{N}$, $\oplus = \max$ et $g = \min$;
- ▷ on cherche un partage dont l'utilité collective est supérieure ou égale à 1.

Tout partage admissible correspond bien entendu à un sous-ensemble $\mathscr{S}'$ de $\mathscr{S}$ de taille inférieure ou égale à K en vertu de la contrainte de volume. De plus, pour tout partage $\overrightarrow{\pi}$, $u_c(\overrightarrow{\pi}) \geq 1$ si et seulement si chaque agent a au moins un objet qu'il désire, c'est-à-dire si $\mathscr{S}'$ contient au moins un objet de chaque sous-ensemble $\mathscr{S}_i$.

Bien entendu, ce résultat s'étend au cas $g = \text{leximin}$. ▲

4.2.5 Contraintes d'exclusion uniquement

Enfin, le dernier cas que nous analysons est celui des contraintes d'exclusion globales, c'est-à-dire s'appliquant à tous les agents et non à des agents particuliers.

Proposition 4.25 (Contraintes d'exclusion globales uniquement) *Le problème [MAX-CUF] restreint aux instances sans autre contrainte que des contraintes d'exclusion globales est* NP-*complet, même dans les cas particuliers où les opérateurs d'agrégation* $(\oplus, g)$ *sont* $(+, (*))$ *ou* $(\max, + | \min | \text{leximin})$.

Démonstration $\oplus = +$ **:** Ce cas peut être traité par une réduction depuis le problème 13, [HITTING SET]. À partir d'une instance $(C = \{\mathscr{S}_1, \ldots, \mathscr{S}_n\}, K)$ de ce problème, nous créons l'instance de [MAX-CUF] suivante :

- ▷ $\mathscr{N} = \{1\}$;
- ▷ $\mathscr{O} = \mathscr{S}$;
- ▷ pour chaque $\mathscr{S}_i \in C$, nous créons une contrainte d'exclusion empêchant l'attribution simultanée de tous les objets de $\mathscr{S}_i$;
- ▷ $\Delta_1 = \{(o, 1) | o \in \mathscr{S}\}$;
- ▷ $\mathscr{V} = \mathbb{N}$, $\oplus = +$ et g est quelconque (il n'y a qu'un agent, donc g ne compte pas) ;
- ▷ on cherche un partage dont l'utilité collective est supérieure ou égale à $|\mathscr{S}| - K$.

Puisqu'un objet au moins de chaque contrainte d'exclusion ne doit pas être attribué, tout partage admissible $\overrightarrow{\pi}$ correspond à un sous-ensemble π_1 des objets de $\mathscr{S}$ tels qu'au moins un objet de chaque $\mathscr{S}_i$ n'est pas attribué, c'est-à-dire que $\mathscr{S} \setminus \pi_1$ est un *hitting set*. Réciproquement, à chaque *hitting set* correspond un partage admissible pour les même raisons. En conséquence, il existe un partage admissible d'utilité collective supérieure ou égale à $|\mathscr{S}| - K$ si et seulement s'il existe un *hitting set* de cardinalité au plus K.

$\oplus = \max$ **et** $g = \min | \text{leximin}$ **:** Nous utilisons ici une réduction directement depuis le problème [SAT] (problème 19). Soit $\varphi = Cl_1 \wedge \cdots \wedge Cl_n$ une formule en forme normale conjonctive (CNF). On crée l'instance de [MAX-CUF] suivante :

- ▷ $\mathscr{N} = \{1, \ldots, n\}$;
- ▷ $\mathscr{O} = \bigcup_{\mathbf{x} \in Var(\varphi)} \{o_\mathbf{x}, \overline{o_\mathbf{x}}\}$;
- ▷ pour tout $\mathbf{x} \in Var(\varphi)$, nous créons une contrainte d'exclusion $\neg(o_\mathbf{x} \wedge \overline{o_\mathbf{x}})$;
- ▷ pour tout i, $\Delta_i = \{(o_\mathbf{x}, 1) | \mathbf{x} \in C_i\} \cup \{(\overline{o_\mathbf{x}}, 1) | \neg\mathbf{x} \in C_i\}$;
- ▷ $\mathscr{V} = \mathbb{N}$, $\oplus = \max$ et $g = \min$;
- ▷ on cherche un partage dont l'utilité collective est supérieure ou égale à 1.

À toute interprétation v de φ correspond un partage admissible $\overrightarrow{\pi}$, car seul un des deux objets $o_\mathbf{x}$ et $\overline{o_\mathbf{x}}$ peut être attribué. Réciproquement, à tout partage admissible tel qu'au moins un des objets $o_\mathbf{x}$ ou $\overline{o_\mathbf{x}}$ est attribué pour tout $\mathbf{x}$ correspond une interprétation de φ

obtenue en instanciant à vrai tous les $\mathbf{x}$ tels que $o_{\mathbf{x}}$ est alloué, et à faux les autres variables. En ce qui concerne les autres partages admissibles $\overrightarrow{\pi}$, on peut toujours les compléter sans faire diminuer l'utilité collective, jusqu'à atteindre un partage tel qu'au moins un des objets $o_{\mathbf{x}}$ ou $\overline{o_{\mathbf{x}}}$ est attribué pour tout $\mathbf{x}$. On peut donc se restreindre à ce dernier type d'allocations.

S'il existe un partage admissible d'utilité supérieure ou égale à 1, alors l'interprétation correspondante satisfait chaque clause de la formule φ, puisque chaque agent i reçoit au moins un objet correspondant à un littéral de la clause i. Réciproquement, s'il existe un modèle de la formule φ, alors il est possible d'attribuer à chaque agent i l'ensemble des objets correspondant aux littéraux satisfaits de la clause : il y en a au moins un par clause, donc l'utilité de chaque agent est 1, ainsi donc que l'utilité collective.

Comme à l'accoutumée, cette réduction fonctionne aussi pour $g =$ leximin.

$\oplus = \max$ **et** $g = +$ **:** Il est relativement facile de vérifier qu'une réduction identique au cas précédent fonctionne, mais cette fois-ci depuis le problème [UNWEIGHTED-MAX-SAT] (problème 22). ▲

4.2.6 Conclusion

Nous nous sommes intéressés dans cette section à la complexité du problème de maximisation de l'utilité collective pour le problème de partage exprimé dans le langage de représentation compacte introduit au chapitre 3. Sans surprise, ce problème est NP-complet, mais ce qui est plus intéressant est qu'il le reste pour la plupart des cas particuliers même très simples pour les opérateurs d'agrégation courants et la limitation des contraintes aux contraintes les plus classiques. De plus, la variété des problèmes NP-complets utilisés pour les réductions dans les preuves montre que le choix des opérateurs et des contraintes conduit à des problèmes de nature relativement différente, même s'ils sont pour la plupart de même complexité. Nous avons toutefois exhibé quelques problèmes polynômiaux, pour des types de demandes très simples (atomiques), des contraintes de préemption uniquement, et certains types d'opérateurs.

Les résultats obtenus sont détaillés dans la figure 4.2. Dans cette figure, les résultats marqués (A) sont des corollaires de résultats déjà connus dans le domaine des enchères combinatoires, les résultats marqués en caractères non gras sont relativement immédiats, le cas marqué « ?» est encore non résolu, et enfin tous les autres résultats, marqués en gras, sont *a priori* non triviaux, et nouveaux à notre connaissance.

[MAX-CUF]

N'importe quel type de contraintes : NP-c.

Pas de contraintes : P

Contraintes d'exclusion seulement

$\oplus = +$, n'importe quel g : NP-c.

$\oplus$ \ g	$+$	(lexi)min
max	**NP-c.**	**NP-c.**

Contraintes de volume

$\oplus = +$, n'importe quel g : NP-c.
$g = +$, n'importe quel $\oplus$: NP-c.
$\oplus = \max$, $g = \min$ ou leximin : NP-c.

Contraintes de préemption

Demandes atomiques

$\oplus$ \ g	$+$	min	leximin
$+$	P (A)	**NP-c., P si poids égaux**	**NP-c.**
max	P	P	?

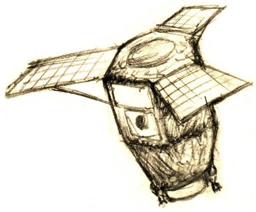
N'importe quel type de demande

$\oplus$ \ g	$+$	(lexi)min
$+$	NP-c. (A)	**NP-c.**
max	**NP-c.**	**NP-c.**

Figure 4.2 — Résumé des résultats de complexité obtenus pour le problème de maximisation de l'utilité collective.

Troisième partie

Algorithmique



Chapitre 5
Préordre leximin et programmation par contraintes

Lors de l'ensemble des chapitres précédents, nous avons introduit entre autres un modèle formel générique permettant d'exprimer des instances du problème de partage de biens indivisibles, et nous avons aussi mis en évidence des résultats de complexité liés à l'expression compacte de ces instances. La partie que nous abordons maintenant a une vocation beaucoup plus pratique. Dans ce chapitre, nous allons nous pencher sur un problème algorithmique particulier pour lequel nous allons proposer plusieurs méthodes de résolution. Ces approches seront testées de manière expérimentale dans le chapitre suivant.

5.1 Exposé du problème

Le problème de maximisation de l'utilité collective dans le modèle de problème de partage que nous avons introduit au chapitre 3 se ramène finalement à un problème d'optimisation combinatoire très classique, pouvant être facilement spécifié et implanté dans un cadre de modélisation et de résolution de problèmes tel que celui de la programmation par contraintes. Cependant, cette traduction s'applique difficilement au préordre leximin, qui est pourtant un ordre de bien-être collectif pertinent dans un contexte de recherche d'équité, comme nous l'avons vu au chapitre 1. Le problème d'optimisation leximin dépasse en outre largement le cadre particulier des problèmes de partage équitable. C'est pourquoi nous avons choisi de nous y intéresser : il s'agit d'un problème algorithmique non trivial, pertinent dans le domaine du partage, et dont la portée dépasse largement ce cadre particulier.

5.1.1 Retour sur le préordre leximin

Le préordre leximin, introduit dans la définition 1.32 page 42 en tant qu'ordre de bien-être collectif, a de nombreuses vertus, que nous rappelons brièvement. Comme il s'agit d'un raffinement efficace de la fonction d'utilité collective égalitariste min, il hérite de toutes les propriétés mathématiques et «éthiques» de cette fonction : anonymat, insensibilité à une dilatation commune croissante quelconque des utilités, juste part garantie. Il vérifie en plus la propriété de réduction des inégalités, l'indépendance vis-à-vis des agents non concernés, et enfin l'unanimité. Son fonctionnement est fondé sur une comparaison successive des utilités des agents, du plus pauvre au plus riche, jusqu'à trouver une différence qui permet alors de discriminer les deux profils : il s'agit donc d'une comparaison lexicographique sur les vecteurs d'utilité triés.

Comme nous l'avons fait remarquer au chapitre 1, le préordre leximin ne se justifie pas exclusivement dans le cadre du partage. Hors de ce domaine, il est aussi employé pour l'agrégation de niveaux de satisfaction de contraintes floues [Dubois *et al.*, 1996; Dubois et Fortemps, 1999; Dubois *et al.*, 2001], ou dans le domaine de l'optimisation multicritère, lorsque l'on cherche à obtenir des solutions relativement bien équilibrées entre les critères, mais cependant Pareto-efficaces.

Notons que l'on fait parfois usage d'un autre raffinement du min que celui apporté par le préordre leximin (et par le préordre discrimin que nous avons évoqué au chapitre 1, mais qui est peu pertinent dans le contexte du partage), l'objectif étant bien sûr toujours de bénéficier des caractéristiques égalitaires de la fonction min tout en garantissant la Pareto-efficacité de la solution trouvée. L'un de ces raffinements classiques, notamment en recherche opérationnelle et en optimisation multicritère, est la méthode max-min / max-sum. Cette méthode consiste à sélectionner l'ensemble des solutions maximisant la somme des utilités (ou des critères) parmi les solutions optimales au sens de la fonction min. Nous pouvons remarquer que si cette solution est séduisante d'un point de vue algorithmique (le calcul d'une solution max-min-optimale et d'une solution optimale au sens de la somme sont des problèmes d'optimisation classiques et que l'on sait bien résoudre), en revanche elle est problématique d'un point de vue «microéconomique», car elle peut éventuellement aboutir à perdre tous les bénéfices de la fonction min, et à se ramener à faire de l'utilitarisme pur. Considérons par exemple un problème pour lequel l'un des agents joue le rôle de fauteur de troubles, et en particulier a des préférences (quasi-)irréalisables. Dans ce cas, cet agent jouera toujours le rôle de l'agent le plus pauvre, et empêchera le fonctionnement normal de la fonction min, en empêchant son action de filtrage des profils inégalitaires, et en laissant l'espace des alternatives min-optimales égal à l'espace des alternatives admissibles en entier. Ensuite, l'agrégateur utilitariste classique somme sera le seul en jeu pour discriminer l'ensemble des profils d'utilité : ce cas se ramène donc à de l'utilitarisme pur. Le préordre leximin, en revanche, permet de résoudre ce cas pathologique en ignorant simplement l'agent fauteur de troubles, et en discriminant les profils d'utilité des autres agents de la même manière que si cet agent particulier n'était pas présent dans le partage.

Ces arguments en faveur du préordre leximin nous incitant à nous intéresser à l'algorithmique qui lui est dédiée, nous allons donc nous pencher sur le problème de recherche d'une solution non dominée au sens du préordre leximin. Il y a deux manières d'aborder cette question :

- ▷ utiliser une fonction d'utilité collective représentant le préordre leximin (ce qui est possible lorsque l'espace des alternatives est fini ou infini dénombrable), et transformer ainsi le problème en optimisation monocritère classique ;
- ▷ aborder le problème directement dans le cadre de l'optimisation multicritère et développer une algorithmique dédiée.

Nous parlerons brièvement de la première approche dans la section 5.4, mais notre point de vue sera plutôt centré sur la vision multicritère du problème.

Notons qu'il existe un certain nombre de travaux récents sur l'optimisation multicritère (ou multiobjectif). Cependant, comme le rappelle [Ehrgott et Gandibleux, 2002], le domaine de l'optimisation et celui de la décision multicritère, tous deux très prolifiques, ont été longtemps séparés. Cela est d'autant plus surprenant que de nombreux problèmes de décision multiobjectif ont naturellement besoin de procédures permettant de calculer un optimum au sens de tous les critères (par exemple un optimum de Pareto). Réciproquement de nombreux problèmes combinatoires réels étudiés dans le cadre de l'optimisation monoobjectif nécessitent la prise en compte de plusieurs critères pour modéliser une application de manière réaliste.

La plupart des travaux dans le domaine de l'optimisation multicritère sont issus de la communauté de la recherche opérationnelle [Ehrgott, 2000], et sont orientés sur des techniques traditionnelles de ce domaine, telles que la programmation linéaire . En ce qui concerne la définition de l'ensemble des bonnes solutions, la plupart de ces travaux se concentrent sur la recherche de l'ensemble des alternatives Pareto-optimales. Plus récemment, certains travaux se sont intéressés plus spécifiquement à des solutions faisant apparaître des compromis entre les critères, telles que les solutions leximin-optimales [Ogryczak, 1997], les moyennes pondérées ordonnées [Ogryczak et Śliwiński, 2003], ou encore la norme de Tchebycheff [Galand et Perny, 2006]. Notons cependant que la plupart de ces travaux sont dédiés à des problèmes particuliers tels que le problème sac-à-dos, celui du voyageur de commerce, ou encore le problème de plus court chemin dans un graphe [Galand et Perny, 2006, 2007; Perny *et al.*, 2007].

Si la prise en compte d'objectifs multiples dans les procédures d'optimisation commence à être étudiée abondamment par les chercheurs opérationnels, en revanche, l'extension du cadre de la programmation par contraintes au multicritère n'en est qu'à ses balbutiements. On peut citer toutefois quelques exceptions, comme par exemple [Rollón et Larrosa, 2006], qui étend l'algorithme classique *Bucket Elimination* au cadre des réseaux de contraintes multiobjectif.

5.2 Le problème de satisfaction de contraintes à critère max-leximin

Notre démarche s'appuie sur le cadre des problèmes de satisfaction de contraintes et de la programmation par contraintes, afin de bénéficier d'un cadre de modélisation et de résolution intuitif et puissant, que nous pourrons exploiter pour l'adapter au problème d'optimisation leximin. Il ne s'agit donc pas de redéfinir le cadre de modélisation et les algorithmes de base de la programmation par contraintes, mais bien d'exploiter pour la résolution de notre problème l'ensemble des outils qu'il nous fournit.

Le cadre de la programmation par contraintes constitue un outil flexible et efficace pour la modélisation et la résolution de problèmes combinatoires tels que la planification, les problèmes d'allocation de ressource ou encore les problèmes de configuration. Ce paradigme est fondé sur la notion de *réseau de contraintes*, que nous avons introduit au chapitre 3, dans la définition 3.3. Rappelons qu'un réseau de contraintes est un triplet $(\mathscr{X}, \mathcal{D}, \mathscr{C})$, où $\mathscr{X} = \{\mathbf{x_1}, \ldots, \mathbf{x_p}\}$ est un ensemble de variables, $\mathcal{D}$ est la fonction de domaine, qui associe à tout $\mathbf{x_i} \in \mathscr{X}$ un domaine fini $\mathscr{D}_{\mathbf{x_i}}$, et $\mathscr{C}$ un ensemble de contraintes spécifiant chacune un ensemble d'instanciations autorisées sur un sous-ensemble des variables.

Le problème central de la programmation par contraintes est le problème de satisfaction de contraintes ([CSP] pour *Constraint Satisfaction Problem*) :

Définition 5.1 (Problème de satisfaction de contraintes ([CSP])) *Le problème de satisfaction de contraintes est le problème de décision suivant :*

Problème 14: [CSP]

INSTANCE : Un réseau de contraintes $(\mathscr{X}, \mathcal{D}, \mathscr{C})$.
QUESTION : $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ *a-t-il une solution (une instanciation complète cohérente) ?*

Comme nous l'avons déjà fait remarquer dans le chapitre 3, il s'agit d'un problème NP-complet.

Il existe une déclinaison de ce problème de satisfaction de contraintes en problème d'optimisation, que nous appellerons problème de satisfaction de contraintes avec variable objectif, et dans lequel une variable du réseau de contraintes, qui prend ses valeurs dans les entiers, doit être maximisée.

Définition 5.2 (Problème des satisfaction de contraintes avec variable objectif) *Le problème de satisfaction de contraintes avec variable objectif est le problème d'optimisation suivant :*

Problème 15: *Problème des satisfaction de contraintes avec variable objectif*

INSTANCE :	*Un réseau de contraintes* $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ *et une variable objectif* $\mathbf{o} \in \mathscr{X}$ *telle que* $\mathscr{D}_\mathbf{o} \subsetneq \mathbb{N}$.
SOLUTION :	*«Inconsistant» si* $sol(\mathscr{X}, \mathcal{D}, \mathscr{C}) = \varnothing$.
	Sinon une solution $\widehat{v} = \operatorname{argmax}_{v \in sol(\mathscr{X}, \mathcal{D}, \mathscr{C})}(v(\mathbf{o}))$.

Étant donnés un réseau de contraintes $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ *et une variable objectif* $\mathbf{o}$*, nous noterons* $max(\mathscr{X}, \mathcal{D}, \mathscr{C}, \mathbf{o})$ *l'ensemble des solutions* $v \in sol(\mathscr{X}, \mathcal{D}, \mathscr{C})$ *telles que* $v(\mathbf{o})$ *est maximal.*

La version décisionnelle de ce problème d'optimisation, pour laquelle on cherche à déterminer s'il existe une solution dont la valeur de la variable objectif est supérieure à un certain seuil K est un problème NP-complet.

Notons que cette déclinaison du problème de satisfaction de contraintes classique en problème d'optimisation est légèrement différente du cadre des problèmes de satisfaction de contraintes valués que nous avons évoqué au chapitre 3, puisqu'ici, la valuation (qui évalue la qualité d'une solution) n'est pas portée par les contraintes elles-mêmes, mais par une variable objectif.

L'expression de problèmes de partage à fonctions d'utilité «classiques» telles que $g^\star$, $g^{(e)}$ ou encore $g^{(N)}$ (introduites dans le chapitre 1) sous forme de problème de satisfaction de contraintes avec variable objectif est relativement immédiate : n variables jouent le rôle des utilités individuelles, une variable joue le rôle de l'utilité collective, et une contrainte lie ces variables entre elles. Ainsi par exemple pour $g^\star$, cette contrainte est simplement $\mathbf{u_c} = \sum_{i=1}^{n} \mathbf{u_i}$.

En revanche, nous devons introduire un nouveau problème d'optimisation pour prendre en compte la multiplicité des variables objectif, et l'utilisation du préordre leximin sur ces variables. La définition suivante introduit la notion de [MAXLEXIMINCSP], qui est adaptée du problème de satisfaction de contraintes avec variable objectif :

Définition 5.3 ([MAXLEXIMINCSP]) *Le problème [MAXLEXIMINCSP] le problème d'optimisation suivant :*

Problème 16: *[MAXLEXIMINCSP]*

INSTANCE :	*Un réseau de contraintes* $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ *et un vecteur objectif* $(\mathbf{u_1}, \ldots, \mathbf{u_n}) \in \mathscr{X}^n$ *tel que* $\forall i,\ \mathscr{D}_{\mathbf{u_i}} \subsetneq \mathbb{N}$.
SOLUTION :	*«Inconsistant» si* $sol(\mathscr{X}, \mathcal{D}, \mathscr{C}) = \varnothing$.
	Sinon une solution $\widehat{v} \in \operatorname{argmax}^{\preceq_{leximin}}_{v \in sol(\mathscr{X}, \mathcal{D}, \mathscr{C})}(v(\mathbf{u_1}), \ldots, v(\mathbf{u_n}))$.

Étant donnés un réseau de contraintes $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ *et un vecteur objectif* $\overrightarrow{\mathbf{u}}$*, nous noterons* $maxleximin(\mathscr{X}, \mathcal{D}, \mathscr{C}, \overrightarrow{\mathbf{u}})$ *l'ensemble des solutions* $v \in sol(\mathscr{X}, \mathcal{D}, \mathscr{C})$ *telles que* $(v(\mathbf{u_1}), \ldots, v(\mathbf{u_n}))$ *est non dominé au sens du préordre leximin, c'est-à-dire* $maxleximin(\mathscr{X}, \mathcal{D}, \mathscr{C}, \overrightarrow{\mathbf{u}}) = \operatorname{argmax}^{\preceq_{leximin}}_{v \in sol(\mathscr{X}, \mathcal{D}, \mathscr{C})}(v(\mathbf{u_1}), \ldots, v(\mathbf{u_n}))$.

La version décisionnelle de ce problème d'optimisation, pour laquelle on cherche à déterminer s'il existe une solution dont la valeur de la variable objectif domine au sens du leximin un certain vecteur $\overrightarrow{K}$ est clairement un problème NP-complet, et ce pour les raisons suivantes. (1) L'appartenance à NP est immédiate, car la comparaison au sens du leximin de deux vecteurs donnés peut être effectuée en temps linéaire. (2) La NP-complétude est un corollaire de la NP-complétude du problème avec variable objectif simple : le [MAXLEXIMINCSP] est équivalent à ce problème si le vecteur objectif est de taille 1.

Nous allons donc nous intéresser à la résolution du problème [MAXLEXIMINCSP]. Notre objectif n'est pas de réinventer les algorithmes de base du domaine de la programmation par contraintes, qui ont été largement étudiés dans la littérature et fonctionnent bien dans le cadre monocritère. Notre but est plutôt d'appliquer les outils fournis par ce cadre de modélisation et de résolution à notre problème d'optimisation multicritère leximin.

Introduisons avant de poursuivre quelques notations supplémentaires. Pour une variable donnée $\mathbf{x}$, nous noterons respectivement $\underline{\mathbf{x}}$ et $\overline{\mathbf{x}}$ pour désigner $\min(\mathscr{D}_\mathbf{x})$ et $\max(\mathscr{D}_\mathbf{x})$. Dans les algorithmes, nous utiliserons aussi les raccourcis suivants pour les réductions de domaines : $\underline{\mathbf{x}} \leftarrow \alpha$ pour désigner une modification de la fonction de domaine $\mathcal{D}$ telle que $\mathscr{D}_\mathbf{x} \leftarrow \mathscr{D}_\mathbf{x} \cap [\![\alpha, +\infty[\![$ (toutes les valeurs inférieures à α sont effacées du domaine de $\mathbf{x}$ — notons que si $\alpha < \underline{\mathbf{x}}$, $\mathscr{D}_\mathbf{x}$ n'est pas modifié), $\overline{\mathbf{x}} \leftarrow \alpha$ pour désigner $\mathscr{D}_\mathbf{x} \leftarrow \mathscr{D}_\mathbf{x} \cap]\!] -\infty, \alpha]\!]$ (toutes les valeurs supérieures à α sont effacées du domaine de $\mathbf{x}$ — notons que si $\alpha > \overline{\mathbf{x}}$, $\mathscr{D}_\mathbf{x}$ n'est pas modifié), et $\mathbf{x} \leftarrow \alpha$ pour désigner $\mathscr{D}_\mathbf{x} \leftarrow \{\alpha\}$ (toutes les valeurs différentes de α sont effacées du domaine de $\mathbf{x}$ — notons que si $\alpha \notin \mathscr{D}_\mathbf{x}$, alors $\mathscr{D}_\mathbf{x}$ devient vide). Pour deux fonctions de domaine $\mathcal{D}_1$ et $\mathcal{D}_2$ sur deux ensembles de variables $\mathscr{X}_1$ et $\mathscr{X}_2$, nous noterons $\langle \mathcal{D}_1, \mathcal{D}_2 \rangle$ pour désigner la fonction de domaine sur $\mathscr{X} = \mathscr{X}_1 \cup \mathscr{X}_2$ telle que $\forall \mathbf{x} \in \mathscr{X}$, $\mathcal{D}(\mathbf{x}) = \mathcal{D}_1(\mathbf{x})$ si $\mathbf{x} \in \mathscr{X}_1$ et $\mathcal{D}(\mathbf{x}) = \mathcal{D}_2(\mathbf{x})$ si $\mathbf{x} \in \mathscr{X}_2$. Enfin, nous pourrons noter une fonction de domaine sous sa forme explicite : $(\mathbf{x_1} : \mathscr{D}_{\mathbf{x_1}}, \dots, \mathbf{x_n} : \mathscr{D}_{\mathbf{x_n}})$.

5.3 Programmation par contraintes et optimisation leximin

L'objectif de cette section est d'introduire des algorithmes de résolution dédiés au problème [MAXLEXIMINCSP], et fondés sur la programmation par contraintes, qui fournit un cadre de résolution dédié aux problèmes de satisfaction de contraintes avec ou sans variable objectif. Avant de nous intéresser aux algorithmes de résolution eux-mêmes, nous allons introduire une petite description du fonctionnement et des principes de base de la programmation par contraintes.

5.3.1 La programmation par contraintes

5.3.1.1 Les deux composantes d'un système de programmation par contraintes

La programmation par contraintes est un paradigme de programmation, issu du domaine des problèmes de satisfaction de contraintes, auxquels se sont intéressés les chercheurs en intelligence artificielle dès les années 70, et né du développement du cadre de la programmation par contraintes logique (CLP pour *Constraint Logic Programming*) dans les années 80. C'est dans ces années-là qu'ont été posées les bases théoriques du cadre sous sa forme actuelle, et qu'ont été développés les premiers systèmes de résolution fondés sur ce paradigme. Les principaux développements qui ont suivi dans les années 90 ont concerné dans un premier temps l'identification de nouveaux champs d'application de la programmation par contraintes, ce qui a conduit à la mise en évidence de nouveaux types de contraintes et à l'introduction des mécanismes de propagation et de filtrage associés. Dans un deuxième temps, un certain nombre d'extensions du problème de satisfaction de contraintes, assorties de leur cadre algorithmique, ont permis de prendre en compte de nouvelles caractéristiques telles que : la notion de préférence avec les problèmes de satisfaction de contraintes valués [Schiex *et al.*, 1995] ou semi-anneaux [Bistarelli *et al.*, 1997] ou encore la distribution avec les problèmes de satisfaction de contraintes distribués [Collin *et al.*, 1992; Faltings, 2006].

Nous allons introduire très rapidement quelques principes fondateurs de la programmation par contraintes. Pour une description plus formelle et plus détaillée de la programmation par contraintes, on pourra se référer par exemple à l'ouvrage de référence [Apt, 2003], à l'article [van Hentenryck *et al.*, 1992], ou à l'introduction du manuel d'OPL Studio [van Hentenryck, 1999]. Pour une vue d'ensemble détaillée et plus générale de la programmation par contraintes, des problèmes de satisfaction de contraintes et de leurs extensions, on pourra consulter l'ouvrage [Rossi *et al.*, 2006].

La programmation par contraintes a pour objet la résolution de problèmes de satisfaction de contraintes, à variables objectif ou non. Ce cadre est construit sur les composantes suivantes [Apt, 2003, chapitre 3] :

▷ *le prétraitement*, dont l'objectif est de transformer le problème initial à traiter en un problème de forme équivalente plus simple que le problème initial, et dans un format accepté par l'algorithme de recherche ;

▷ une procédure d'*exploration de l'arbre de recherche*, elle-même fondée sur :
  - une procédure de branchement, dont le rôle est de séparer le problème à traiter en deux sous-problèmes complémentaires (plus simples), si cela est possible, en choisissant par exemple une variable à instancier à une valeur donnée (dans le cas d'un algorithme de type retour-arrière ou *branch-and-bound* par exemple),
  - une condition permettant de tester si le problème en cours est atomique ou s'il peut encore être traité par la procédure de branchement,
  - une condition d'arrêt de la procédure d'exploration ;

▷ une procédure de *propagation de contraintes*, dont le rôle est de transformer le problème en cours en un problème équivalent plus simple, par déduction d'informations selon l'affectation des variables en cours et la nature des contraintes du problème.

Si l'on met à part la procédure de prétraitement, qui occupe une place relativement marginale dans la résolution d'un problème, un système de programmation par contraintes est donc fondé sur deux composantes principales : celle qui concerne l'exploration de l'arbre de recherche, et celle qui concerne la propagation de contraintes. Le processus de résolution d'un problème de satisfaction de contraintes résulte d'un dialogue permanent entre ces deux composantes fondamentales, comme l'illustre la figure 5.1.

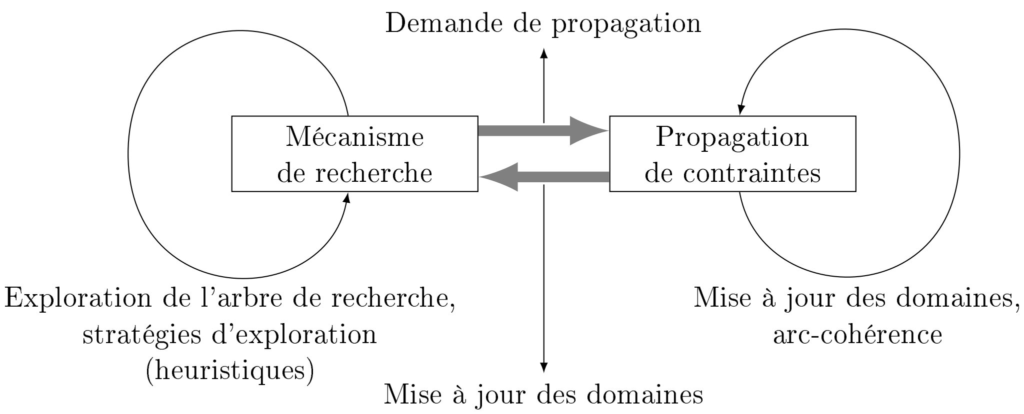


Figure 5.1 — Illustration simplifiée du principe de la programmation par contraintes.

Dans un système de programmation par contraintes, l'utilisateur a généralement assez peu d'emprise sur la composante d'exploration de l'arbre de recherche. L'algorithme de résolution (correspondant aux fonctions **solve** et **maximize** que nous allons introduire plus loin) est en général fixé par le système, et l'action de l'utilisateur se résume à spécifier un ensemble d'heuristiques permettant de guider la procédure de branchement dans son exploration de l'arbre de recherche en lui indiquant par exemple la prochaine variable à instancier, et la prochaine valeur à choisir dans son domaine.

En ce qui concerne la deuxième composante fondamentale d'un système de programmation par contraintes, celle-ci repose sur un ensemble de mécanismes de propagation d'information et de filtrage associés à un ensemble de contraintes. Cette composante comporte un langage de spécification des contraintes permettant la description du problème à traiter, et un ensemble d'algorithmes de filtrage associés aux contraintes. Certains langages tels que Choco [Laburthe, 2000] permettent l'introduction de nouvelles contraintes dans le système, grâce à la spécification d'algorithmes de filtrage dédiés. L'objectif de ces algorithmes de filtrage est d'exploiter la sémantique des contraintes afin de tâcher d'éliminer au plus tôt possible (et de manière la plus efficace possible en terme de temps de calcul) les valeurs des domaines des variables qui ne font pas partie d'instanciations cohérentes du problème. Bien entendu, cette notion de filtrage est toujours fondée sur un compromis entre le temps de calcul nécessaire à la propagation de contraintes et la quantité d'information déduite.

Nous allons introduire de manière plus détaillée comment cette notion de filtrage a été formalisée, et comment elle a été implantée en programmation par contraintes.

5.3.1.2 Propagation de contraintes

La propagation de contraintes est l'une des techniques les mieux formalisées, les plus efficaces, et les plus abouties pour faire de l'inférence dans les réseaux de contraintes. Son objectif est de détecter et d'éliminer au plus tôt les éléments du problème qui n'ont aucune influence sur la consistance du réseau de contraintes. En d'autres termes, on cherche à transformer le réseau de contraintes initial en un réseau qui soit plus simple, et en même temps équivalent (en termes de solutions) au réseau initial, soit par suppression de contraintes qui sont toujours vérifiées (notion d'*entailement*), soit par suppression de valeurs des domaines des variables si ces valeurs ne peuvent pas faire partie d'une instanciation cohérente. Prenons un exemple. Soit un réseau de contraintes constitué de 2 variables $\mathbf{x_1}$ et $\mathbf{x_2}$ de même domaine $\{1, 2, 3\}$, et d'une contrainte $\mathbf{x_1} < \mathbf{x_2}$. À l'aide d'un raisonnement basique on peut s'apercevoir que $\mathbf{x_1}$ ne peut pas prendre la valeur 3, et que $\mathbf{x_2}$ ne peut pas prendre la valeur 1, à cause de la contrainte $\mathbf{x_1} < \mathbf{x_2}$. Nous pouvons donc supprimer ces valeurs des domaines de $\mathbf{x_1}$ et $\mathbf{x_2}$ sans restreindre l'ensemble des solutions du réseau. Tenir un tel raisonnement revient à effectuer une tâche de propagation de contraintes. On pourra trouver une description formelle de ces notions dans [Bessière, 2006].

La notion la plus classique du domaine de la propagation de contraintes est la notion d'*arc-cohérence*. Cette notion est due initialement à [Mackworth, 1977a], qui l'a définie dans un premier temps dans le cadre des contraintes binaires, puis l'a étendue aux contraintes n-aires.

Définition 5.4 (Support et arc-cohérence généralisée) *Soit $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ un réseau de contraintes, $C \in \mathscr{C}$ tel que $\mathscr{X}(C) = \{\mathbf{x_1}, \ldots, \mathbf{x_k}\}$, et $\alpha \in \mathscr{D}_{\mathbf{x_i}}$. Le* support *de α pour $\mathbf{x_i}$ et pour la contrainte C est l'ensemble des instanciations $v \in \mathscr{X}(C)$ telles que $v(\mathbf{x_i}) = \alpha$ et $v \in \mathscr{R}(C)$.*

C est dite arc-cohérente généralisée *(ou gac) si et seulement si $\forall \mathbf{x} \in \mathscr{X}(C)$, $\forall \alpha \in \mathscr{D}_{\mathbf{x}}$, α a un support non vide pour $\mathbf{x}$ et la contrainte C.*

$(\mathscr{X}, \mathcal{D}, \mathscr{C})$ est arc-cohérent généralisé *si et seulement si toutes ses contraintes le sont.*

Cette notion d'arc-cohérence généralisée est intuitive : une valeur d'un domaine qui n'a pas de support sur une contrainte ne peut conduire à une solution du réseau de contraintes. Le terme «généralisé» associé à l'arc-cohérence est dû au fait qu'historiquement cette notion a été introduite pour des contraintes binaires uniquement.

Nous pouvons définir, pour un réseau de contraintes donné potentiellement non arc-cohérent généralisé, un réseau de contraintes arc-cohérent équivalent :

Définition 5.5 (fonction *gac*) *Soit $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ un réseau de contraintes consistant et $C \in \mathscr{C}$ une contrainte. Nous noterons $gac(\mathscr{X}, \mathcal{D}, \mathscr{C}, C)$ le réseau de contraintes $(\mathscr{X}, \mathcal{D}', \mathscr{C})$, avec $\mathcal{D}'$ la fonction de domaine telle que pour tout i $\mathcal{D}(\mathbf{x_i})'$ est le sous-ensemble de $\mathcal{D}(\mathbf{x_i})$ de cardinalité maximale tel que tout $\alpha \in \mathcal{D}(\mathbf{x_i})'$ a un support non vide pour C.*

Nous noterons de même $gac(\mathscr{X}, \mathcal{D}, \mathscr{C})$ le réseau de contraintes $(\mathscr{X}, \mathcal{D}', \mathscr{C})$, avec $\mathcal{D}'$ la fonction de domaine telle que pour tout i $\mathcal{D}(\mathbf{x_i})'$ est le sous-ensemble de $\mathcal{D}(\mathbf{x_i})$ de cardinalité maximale tel que tout $\alpha \in \mathcal{D}(\mathbf{x_i})'$ a un support non vide pour toute contrainte de $\mathscr{C}$.

Nous pouvons vérifier la validité de cette définition. Si le réseau de contraintes est consistant, alors chaque domaine contient au moins une valeur dont le support est non vide pour chaque contrainte ; donc il existe au moins un réseau de contraintes arc-cohérent généralisé inclus dans le réseau initial. De plus, pour une variable donnée $\mathbf{x_i}$, il ne peut y avoir deux sous-ensembles différents $\mathscr{D}'_{\mathbf{x_i}}$ et $\mathscr{D}''_{\mathbf{x_i}}$, tous deux inclus dans $\mathscr{D}_{\mathbf{x_i}}$, de cardinalité maximale, et tels que tout $\alpha \in \mathscr{D}'_{\mathbf{x_i}}$ et tout $\alpha \in \mathscr{D}''_{\mathbf{x_i}}$ ont un support non vide pour toute contrainte de $\mathscr{C}$. Donc la fonction $gac(\mathscr{X}, \mathcal{D}, \mathscr{C})$ définit bien un réseau de contraintes unique.

La question de la complexité du calcul de $gac(\mathscr{X}, \mathcal{D}, \mathscr{C})$, ou plus particulièrement de $gac(\mathscr{X}, \mathcal{D}, \mathscr{C}, C)$ pour une contrainte donnée est fondamentale, car il s'agit d'une procédure de base de la propagation de contraintes.

La première procédure de calcul d'arc-cohérence sur un réseau de contraintes a été introduite dans [Mackworth, 1977a], restreint aux contraintes binaires (algorithme AC3), et dans [Mackworth, 1977b] pour les contraintes n-aires (algorithme GAC3). Cette procédure permet le calcul de l'arc-cohérence généralisée sur un réseau de contraintes en temps $O(er^3d^{r+1})$ et en espace $O(er)$, où e est le nombre de contraintes du réseau, r désigne la plus grande arité parmi les contraintes, et d la taille du plus grand domaine des variables. Depuis, de nombreux travaux dédiés à l'arc-cohérence (généralisée) ont conduit à l'introduction d'un certain nombre d'algorithmes plus performants : (G)AC4 [Mohr et Masini, 1998], AC6 [Bessière et Cordier, 1993], ou AC2001 [Bessière et Régin, 2001]. On pourra trouver une étude détaillée sur l'identification et la complexité de problèmes liés à l'arc-cohérence généralisée dans [Bessière *et al.*, 2007].

Il existe d'autres notions de cohérence dans les réseaux de contraintes. Certaines sont plus fortes que l'arc-cohérence généralisée, comme par exemple la chemin-cohérence [Montanari, 1974] — ou de manière plus générale la k-cohérence [Freuder, 1982] — ou la singleton-arc-cohérence. Ces propriétés permettent donc d'effectuer plus de réductions sur les domaines des variables au prix d'une complexité plus grande.

Il existe aussi des propriétés de cohérence plus faibles que l'arc-cohérence généralisée. Nous nous intéresserons plus particulièrement à l'une d'entre elles, la *borne-cohérence*, définie pour des variables ayant des domaines entiers (ou de manière plus générale totalement ordonnés) :

Définition 5.6 (Borne-cohérence) *Soit $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ un réseau de contraintes et $C \in \mathscr{C}$ telle que $\forall \mathbf{x} \in \mathscr{X}(C)$, $\mathscr{D}_{\mathbf{x}} \subsetneq \mathbb{N}$. C est* borne-cohérente *(ou bc) si et seulement si $\forall \mathbf{x} \in \mathscr{X}(C)$, le support de $\underline{\mathbf{x}}$ et de $\overline{\mathbf{x}}$ sont non vides.*

Nous pouvons définir, à l'instar de l'arc-cohérence généralisée la fonction bc, qui transforme un réseau de contraintes en réseau borne-cohérent.

Définition 5.7 (fonction *bc*) *Soit $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ un réseau de contraintes consistant et $C \in \mathscr{C}$ une contrainte. Nous noterons $bc(\mathscr{X}, \mathscr{D}, \mathscr{C}, C)$ le réseau de contraintes $(\mathscr{X}, \mathscr{D}', \mathscr{C})$, avec $\mathcal{D}'$ la fonction de domaine telle que pour tout i $\mathcal{D}(x_i)'$ est le sous-ensemble de $\mathcal{D}(x_i)$ de cardinalité maximale tel que $\min(\mathscr{D}_{\mathbf{x_i}})$ et $\max(\mathscr{D}_{\mathbf{x_i}})$ ont un support non vide pour C.*

Nous noterons de même $bc(\mathscr{X}, \mathcal{D}, \mathscr{C})$ le réseau de contraintes $(\mathscr{X}, \mathcal{D}', \mathscr{C})$, avec $\mathcal{D}'$ la fonction de domaine telle que pour tout i $\mathcal{D}(x_i)'$ est le sous-ensemble de $\mathcal{D}(x_i)$ de cardinalité maximale tel que tout $\min(\mathscr{D}_{\mathbf{x_i}})$ et $\max(\mathscr{D}_{\mathbf{x_i}})$ ont un support non vide pour toute contrainte de $\mathscr{C}$.

Calculer la borne-cohérence sur un réseau de contraintes donné permet dans certains cas de simplifier la procédure de filtrage (au détriment bien entendu de la quantité d'information inférée), comme nous le verrons lors de l'introduction des contraintes globales.

5.3.1.3 Contraintes globales

Les travaux de ces quinze dernières années dans le domaine de la programmation par contraintes ont mis en évidence l'existence d'un certain nombre de «schémas» de contraintes spécifiques que l'on retrouve dans la modélisation d'un grand nombre de problèmes réels très différents. Les contraintes ainsi définies ont une arité variant avec le problème, mais une sémantique précise et commune à toutes les applications : de telles contraintes sont appelées *contraintes globales*. L'exemple le plus connu est certainement la contrainte globale **AllDifferent**, portant sur un ensemble de variables et interdisant l'instanciation de deux de ces variables à la même valeur. La puissance expressive et l'efficacité opérationnelle des algorithmes de propagation dédiés aux contraintes globales en ont fait un sujet de prédilection de la littérature de ces dernières années en matière de programmation par contraintes. On pourra trouver en particulier sur le sujet une réflexion sur la notion de globalité d'une contrainte dans [Bessière et van Hentenryck, 2003], centrée autour des concepts de globalité sémantique, globalité opérationnelle, et globalité algorithmique. On trouvera aussi une liste étendue de contraintes globales dans [Beldiceanu *et al.*, 2005], assortie d'un état de l'art sur le sujet dans [Beldiceanu *et al.*, 2007].

L'inconvénient des procédures de filtrage par arc-cohérence introduites ci-avant est leur complexité temporelle dépendant exponentiellement de l'arité des contraintes : il est inenvisageable d'appliquer ces procédures de filtrage sur des contraintes globales, qui peuvent avoir une arité importante. Deux stratégies sont envisageables pour contourner ce problème :

- ▷ décomposer les contraintes globales en un ensemble équivalent de contraintes d'arité inférieure (binaires si possible) ;
- ▷ développer des algorithmes de filtrage spécifiques, qui exploitent la sémantique des contraintes globales pour assurer l'arc-cohérence généralisée (ou la borne-cohérence) en temps raisonnable.

Ces deux approches sont parfaitement illustrées par les algorithmes dédiés au problème [MAXLEXIMINCSP] que nous allons introduire dans ce chapitre. Par exemple, l'algorithme 5, que nous allons présenter dans la section 5.4.3.2 fait usage d'une contrainte globale de tri d'un vecteur de variables, pour laquelle une procédure de filtrage efficace a été introduite dans [Bleuzen-Guernalec et Colmerauer, 1997] et dans [Mehlhorn et Thiel, 2000]. L'algorithme 8 que nous allons introduire dans la section 5.4.3.4 est en revanche fondé sur une décomposition de la contrainte **Sort** en contraintes **Max** et **Min** d'arité 3.

5.3.1.4 Programmation par contraintes événementielle

Pour la plupart des procédures de filtrage par arc-cohérence généralisée ou borne-cohérence, il est possible de tirer parti des propagations précédentes pour épargner du travail de vérification de support inutile. Par exemple, supposons qu'à une certaine étape, la contrainte linéaire $\mathbf{x} \geq \mathbf{y}$ soit arc-cohérente. Si la borne supérieure $\overline{\mathbf{y}}$ est réduite, alors il est inutile d'établir à nouveau l'arc-cohérence sur cette contrainte. En revanche, si c'est la borne inférieure $\underline{\mathbf{y}}$ qui est augmentée, alors certaines valeurs de $\mathbf{x}$ peuvent ne plus avoir de support, mais l'établissement de l'arc-cohérence se limite ici à supprimer les valeurs de $\mathbf{x}$ strictement inférieures à $\underline{\mathbf{y}}$.

Certains systèmes de programmation par contraintes actuels (en particulier le système Choco [Laburthe, 2000] que nous avons utilisé pour les expérimentations) traduisent cette prise en compte des propagations précédentes grâce à un système de programmation événementielle. Concrètement, ce système fonctionne par dialogue entre la procédure d'exploration de l'arbre de recherche, qui se charge d'instancier les variables, et les procédures de propagation de contraintes. Celles-ci sont invoquées uniquement si cela est nécessaire, c'est-à-dire si le domaine d'une de leurs variables a été modifié (soit par la procédure d'exploration de l'arbre de recherche elle-même, soit par une propagation de contraintes antérieure). Ces procédures peuvent être déclenchées principalement par quatre types d'évènements :

1. augmentation de la borne inférieure d'une variable ;
2. diminution de la borne supérieure d'une variable ;
3. instanciation d'une variable (combinaison des deux premiers évènements) ;
4. effacement d'une valeur du domaine d'une variable.

Chaque fois que l'un de ces évènements survient, il est ajouté dans une file afin d'être traité dès que possible. Le traitement des évènements consiste à prévenir chaque contrainte concernée par l'évènement d'effectuer les filtrages nécessaires. Chaque contrainte peut réagir différemment à chaque type d'évènement et peut éventuellement déclencher d'autres évènements si le filtrage réduit les domaines des variables. La taille finie des domaines nous assure que cette procédure se termine (car un évènement n'est déclenché qu'en cas de réduction stricte d'un domaine, donc il ne peut y avoir un nombre infini d'évènements).

La plupart des systèmes de programmation par contraintes mettent à disposition un ensemble de contraintes classiquement utilisées, accompagnées de leurs algorithmes de filtrage associés aux évènements cités ci-dessous. Il est cependant possible dans certaines implantations de ces systèmes (comme par exemple Choco) de spécifier ses propres contraintes dotées d'algorithmes de filtrage spécifiques.

Dans la suite de ce chapitre, nous considérons que nous avons à notre disposition un système de programmation par contraintes autorisant la définition de contraintes spécifiques, et doté des deux fonctions suivantes (que nous utiliserons comme des boîtes noires) :

- ▷ **solve**$(\mathscr{X}, \mathcal{D}, \mathscr{C})$, qui renvoie une solution v du réseau de contraintes s'il y en a une, et «Inconsistant» sinon ;
- ▷ **maximize**$(\mathscr{X}, \mathcal{D}, \mathscr{C}, \mathbf{y})$, qui renvoie une solution du problème de satisfaction de contraintes avec variable objectif prenant en entrée $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ et $\mathbf{y}$.

Nous supposerons — contrairement à la plupart des solveurs de contraintes réels — que ces fonctions ne modifient pas le réseau de contraintes en entrée. Cette hypothèse est importante pour la définition des algorithmes introduits dans ce chapitre.

Équipés de ces deux fonctions, et de la possibilité d'utiliser des contraintes globales existantes et d'en définir des nouvelles, nous allons nous atteler à la résolution du problème [MAXLEXIMINCSP].

5.4 Algorithmes de programmation par contraintes

La recherche d'une solution leximin-optimale n'est pas un problème algorithmique très compliqué lorsque le réseau de contraintes en entrée n'a que très peu de solutions : puisqu'une solution leximin-optimale est aussi une solution maximin-optimale, on peut envisager de calculer toutes les solutions maximin-optimales (problème qui peut être modélisé comme un problème de satisfaction de contraintes à variable objectif unique) et de les comparer entre elles afin d'en trouver une leximin-optimale. Cette solution est suggérée par exemple dans [Ehrgott, 2000, p. 162], et, bien qu'elle puisse paraître relativement naïve, elle peut s'avérer efficace sur certaines classes de problèmes. En conséquence, ainsi que le suggère [Ogryczak, 1997], elle ne doit pas être complètement mise de côté.

Cependant, certaines instances possèdent un nombre démesuré de solutions maximin, et ainsi nécessitent une approche légèrement plus astucieuse pour envisager leur résolution en un temps raisonnable. Les aspects algorithmiques liés au calcul de solutions leximin-optimales ont été traités dans plusieurs travaux issus de plusieurs communautés différentes. Tout d'abord, les chercheurs opérationnels s'intéressent aux solutions leximin-optimales dans un contexte d'optimisation multicritère (voir par exemple [Ehrgott, 2000]) : leur domaine d'application concerne par exemple les problèmes d'allocation équitable de ressource [Luss, 1999], les problèmes de répartition d'infrastructures [Ogryczak, 1997], ou encore les jeux matriciels [Potters et Tijs, 1992].

Si le préordre leximin est largement étudié dans la communauté de la théorie de la décision et en recherche opérationnelle, il suscite aussi l'intérêt dans le domaine des problèmes de satisfaction de contraintes flexibles, où il apparaît comme un opérateur d'agrégation de niveaux de satisfaction de contraintes floues pertinent [Dubois *et al.*, 1996; Dubois et Fortemps, 1999; Dubois *et al.*, 2001].

Si les algorithmes dédiés au calcul de solutions leximin-optimales ont donc été naturellement étudiés dans les domaines cités ci-dessus, ils n'ont en revanche jamais été traduits, à notre connaissance, dans le cadre de la programmation par contraintes. Notre première contribution sur le sujet concerne donc l'adaptation des algorithmes existants à ce cadre de modélisation et de résolution, adaptation fondée sur l'introduction :

1. d'algorithmes génériques de calcul de solutions leximin-optimales utilisant les fonctions **solve** et **maximize**, introduites ci-avant comme des «boîtes noires» fournies par les solveurs de contraintes ;

2. d'algorithmes de propagation de contraintes pour toutes les contraintes globales nécessaires utilisées dans les algorithmes introduits.

En outre, les algorithmes présentés dans les travaux évoqués ci-dessus ont souvent un champ d'application limité à des problèmes réalistes mais faciles (par exemple des problèmes continus avec des fonctions objectif linéaires, ou au moins convexes), ou peuvent rapidement devenir déraisonnables en pratique dans certains cas, comme nous allons le voir un peu plus loin dans cette section. Une exception cependant concerne le travail présenté dans [Ogryczak, 1997] et citant l'article [Maschler *et al.*, 1992], qui présente brièvement un algorithme efficace pour le calcul de solutions leximin-optimales dans le cas discret. Ce travail est à la base de l'algorithme 8 que nous présentons en section 5.4.3.4.

Notre seconde contribution sur le sujet concerne l'introduction de plusieurs algorithmes nouveaux qui s'appuient sur la puissance du cadre de la programmation par contraintes pour calculer des solutions leximin-optimales de différentes manières. La plupart de ces algorithmes sont fondés sur des mécanismes de propagation de contraintes existant dans la littérature et adaptés à notre problème.

Afin d'illustrer la manière dont fonctionnent les algorithmes, nous utiliserons l'exemple suivant qui sera décliné tout au long de la section :

Exemple 5.1 Soit $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ un réseau de contraintes, et soit $(\mathbf{u_1}, \mathbf{u_2}, \mathbf{u_3}) \in \mathscr{X}^3$ un vecteur objectif. Nous supposons que l'ensemble des solutions du réseau de contraintes conduit à l'ensemble suivant de valeurs possibles pour le vecteur objectif : $(1,1,0)$, $(5,5,3)$, $(7,3,5)$, $(1,2,1)$, $(9,5,2)$, $(3,4,3)$, $(5,3,6)$ et $(10,3,4)$. Notons que cette instance possède 5 solutions maximin-optimales différentes, qui sont $(5,5,3)$, $(7,3,5)$, $(3,4,3)$, $(5,3,6)$ et $(10,3,4)$, et seulement une solution leximin-optimale, qui est $(7,3,5)$. On peut aussi remarquer que cette dernière solution est différente de la solution maximisant la somme des composantes du vecteur objectif (correspondant au point de vue utilitariste classique), qui est $(10,3,4)$.

5.4.1 Le leximin comme une fonction d'utilité collective

Intéressons-nous dans un premier temps à la résolution du problème [MAXLEXIMINCSP] par encodage du préordre leximin sous la forme d'une fonction d'utilité collective. Le but de cette approche est comme nous l'avons vu de se ramener à un problème d'optimisation monocritère, pour lequel il existe des algorithmes de résolution efficaces. On cherche donc à analyser l'efficacité pratique de la résolution du [MAXLEXIMINCSP] par l'introduction d'une variable objectif $\mathbf{u_c}$ représentant l'utilité collective, et liée au vecteur objectif initial $\overrightarrow{\mathbf{u}}$ par une contrainte représentant la fonction d'utilité collective.

Le premier problème concerne la taille du domaine de $\mathbf{u_c}$, à cause de la nature combinatoire de l'espace des alternatives. Si nous supposons que tous les $\mathscr{D}_{\mathbf{u_i}}$ sont identiques — ce qui constitue une hypothèse raisonnable, car pour que le préordre leximin soit pertinent, les composantes du vecteur objectif doivent être exprimées sur une échelle commune — et de taille m, alors on peut prouver que le nombre de classes d'équivalence pour le préordre leximin sur $\overrightarrow{\mathbf{u}}$ (correspondant à la taille minimale de $\mathscr{D}_{\mathbf{u_c}}$) est :

$$\binom{m+n-1}{n} = \frac{(m+n-1)!}{(m-1)!n!}.$$

La preuve de ce résultat (voir par exemple [Knuth, 1968, exercice 1.2.6–60]) est rappelée dans l'annexe B. On montre aisément (voir la preuve dans la même annexe) que ce nombre est équivalent à m^n lorsque $m \to \infty$. Cela peut rapidement devenir un problème, lorsque m augmente, car la plupart des systèmes de programmation par contraintes ont des difficultés à prendre en compte et à traiter des domaines énormes de manière efficace.

Outre l'explosion de la taille du domaine, l'encodage du leximin par une fonction d'utilité collective pose un deuxième problème, lié à la manière de spécifier la fonction d'utilité collective par une contrainte entre $\mathbf{u_c}$ et $\overrightarrow{\mathbf{u}}$. Trois fonctions d'utilité collective représentant le préordre leximin sont connues :

- ▷ $g_1 : \overrightarrow{x} \mapsto -\sum_{i=i}^{n} n^{-x_i}$ (adapté d'une remarque dans [Frisch *et al.*, 2003]) ;
- ▷ $g_2 : \overrightarrow{x} \mapsto -\sum_{i=1}^{n} x_i^{-q}$, où $q > 0$ est assez grand [Moulin, 1988] (la détermination de l'indice q minimal tel que cette fonction représente l'ordre leximin ne semble pas une question facile à élucider — cette question est étudiée en détails dans l'annexe B).
- ▷ Une fonction moyenne pondérée ordonnée [Yager, 1988] $g_3 : \overrightarrow{x} \mapsto \sum_{i=i}^{n} w_i \cdot u_i^{\uparrow}$, où $w_1 \gg w_2 \gg \cdots \gg w_n$ (où $x \gg y$ signifie de manière informelle «x est beaucoup plus grand que y», voir aussi l'annexe B).

Dans le cas général, ni la contrainte $\mathbf{u_c} = -\sum_{i=i}^{n} n^{-\mathbf{u_i}}$, ni la contrainte $\mathbf{u_c} = -\sum_{i=1}^{n} \mathbf{u_i}^{-q}$ ne sont faciles à propager. Pour ce qui est de la contrainte $\mathbf{u_c} = \sum_{i=i}^{n} w_i \cdot u_i^{\uparrow}$ (fondée sur une moyenne pondérée ordonnée), nous pouvons remarquer que sa propagation est quasiment équivalente à la propagation d'une contrainte de tri sur le vecteur objectif. Si nous sommes en mesure de propager correctement cette contrainte, alors nous pouvons employer directement l'algorithme fondé sur la contrainte **Sort**, que nous allons introduire dans la section 5.4.3.2. L'utilisation d'une moyenne pondérée ordonnée s'avère donc inutile.

Ces remarques semblent donc dissuasives pour l'utilisation de cette méthode pour le calcul d'une solution leximin-optimale. Nous nous devons cependant de nuancer ce propos. Dans certains cas, cette approche peut s'avérer efficace — du moins en théorie. Par exemple, considérons le cas d'un problème d'allocation de ressource multiagent, pour lequel on doit attribuer un et un seul objet à chaque agent, l'allocation de l'objet j à l'agent i produisant l'utilité $u_i = w_{ij}$. Dans ce cas, l'utilité collective peut être calculée de manière linéaire, si l'on effectue une dilatation des poids w_{ij} préalable à la résolution du problème : chaque poids w_{ij} est dilaté en $-w_{ij}^{-q}$. Le calcul des utilités individuelles se fait par une simple contrainte linéaire sur les variables de décision représentant l'allocation des objets aux agents, et le calcul de l'utilité collective par une simple somme sur les utilités des agents. Bien entendu, si le problème de propagation de la contrainte permettant le calcul de l'utilité collective est résolu dans ce cas précis, en revanche, le problème d'explosion du domaine de la variable $\mathbf{u_c}$ subsiste.

Voici un exemple illustrant la manière dont peut être traduit le préordre leximin sous la forme d'une fonction d'utilité collective :

Exemple 5.1.a Dans l'exemple 5.1 donné en début de section, la fonction d'utilité collective définie par $u_c : (u_1, u_2, u_3) \mapsto -(u_1+1)^{-9} - (u_2+1)^{-9} - (u_3+1)^{-9}$ est adéquate pour la représentation du préordre leximin. Le choix de l'exposant a été calculé de manière numérique (dichotomique) à l'aide de l'équation B.3 présentée en annexe B. Le remplacement de u_i par $u_i + 1$ dans le calcul de l'utilité collective nous empêche d'être en dehors du domaine de définition de la fonction u_c (définie pour $u_i > 0$). Les valeurs des utilités collectives des solutions admissibles sont approximativement les suivantes : $u_c(1,1,0) = -1.00$, $u_c(5,5,3) = -4.01 \times 10^{-6}$, $u_c(7,3,5) = -3.92 \times 10^{-6}$, $u_c(1,2,1) = -3.96 \times 10^{-3}$, $u_c(9,5,2) = -5.09 \times 10^{-5}$, $u_c(3,4,3) = -8.14 \times 10^{-6}$, $u_c(5,3,6) = -3.94 \times 10^{-6}$ et $u_c(10,3,4) = -4.33 \times 10^{-6}$. Nous pouvons vérifier que le vecteur leximin-optimal $(7,3,5)$ est celui qui a l'utilité collective la plus élevée.

Notre opinion est que la résolution du problème d'optimisation leximin par l'introduction d'une fonction d'utilité collective pose non seulement les problèmes décrits ci-avant, mais en plus dissimule la véritable sémantique du leximin, et nous empêche de tirer partie de cette sémantique dans la résolution du problème d'optimisation. Les algorithmes que nous allons présenter par la suite s'appuieront donc sur une approche directe du problème, dans le cadre multicritère.

5.4.2 Une contrainte *ad-hoc* pour l'ordre leximin

Le premier algorithme que nous présentons ici est inspiré du principe du *branch-and-bound* (ou séparation-évaluation) pour la résolution de problèmes de satisfaction de contraintes valués par exemple. Le principe du *branch-and-bound* pour les problèmes de maximisation est fondé sur le maintien à chaque nœud de l'arbre de recherche de deux valeurs : une borne inférieure de la valeur objectif, qui correspond à une sous-estimation de la valeur optimale, actualisée à chaque fois qu'une solution est trouvée, et une borne supérieure de la valeur objectif, qui est une sur-estimation de la valeur objectif étant donné le nœud courant de l'arbre de recherche. À chaque nœud, la borne supérieure courante ub est comparée à la borne inférieure lb. Si $ub < lb$, alors la branche en cours d'exploration ne peut conduire à une solution optimale, et il est ainsi inutile de continuer à l'explorer (on élague la branche). La performance de l'algorithme dépend directement de notre capacité à trouver de bons majorants, et donc de notre puissance d'élagage.

Le principe du *branch-and-bound* est très facilement transposable au problème [MAXLEXIMINCSP], car le leximin définit un préordre total sur l'ensemble des tuples possibles du vecteur objectif. L'article [Fargier *et al.*, 2004b] propose une extension directe du *branch-and-bound* à n'importe quel ordre social, donc *a fortiori* au leximin. La procédure que nous présentons ici (algorithme 1) est une transcription un peu plus indirecte, faisant usage des outils mis à disposition par la programmation par contraintes.

Dans l'algorithme 1, les notions de base du *branch-and-bound* apparaissent de manière cachée. L'appel à **solve** de la ligne 5 correspond à l'exploration de l'arbre de recherche pour trouver une meilleure solution que la solution courante (nous supposons que la procédure **solve** s'arrête à la première solution trouvée). Ce qui joue le rôle de l'élagage et ainsi empêche la fonction **solve** d'explorer des branches non-optimales et de renvoyer une solution non optimale est le filtrage associé à la contrainte **Leximin** introduite à la ligne 4 dont voici la définition :

Algorithme 1 — Résolution du problème [MAXLEXIMINCSP] à la manière *branch-and-bound*.

entrée : Un réseau de contraintes $(\mathscr{X}, \mathcal{D}, \mathscr{C})$; un vecteur objectif $(\mathbf{u_1}, \ldots, \mathbf{u_n}) \in \mathscr{X}^n$.
sortie : Une solution au problème [MAXLEXIMINCSP].

1 $\widehat{v} \leftarrow$ **null** ; $v \leftarrow$ **solve**$(\mathscr{X}, \mathcal{D}, \mathscr{C})$;
2 **tant que** $v \neq$ *«Inconsistant»* **faire**
3 | $\widehat{v} \leftarrow v$;
4 | $\mathscr{C} \leftarrow \mathscr{C} \cup \{$**Leximin**$(\widehat{v}(\overrightarrow{\mathbf{u}}), \overrightarrow{\mathbf{u}})\}$;
5 | $v \leftarrow$ **solve**$(\mathscr{X}, \mathcal{D}, \mathscr{C})$;
6 **si** $\widehat{v} \neq$ **null alors retourner** $\widehat{v}$ **sinon retourner** *«Inconsistant»*;

Définition 5.8 (Contrainte Leximin) *Soient $\overrightarrow{\mathbf{x}}$ un vecteur de variables, $\overrightarrow{\lambda}$ un vecteur d'entiers, et v une instanciation. La contrainte* **Leximin**$(\overrightarrow{\lambda}, \overrightarrow{\mathbf{x}})$ *porte sur l'ensemble de variables de $\overrightarrow{\mathbf{x}}$, et est satisfaite par v si et seulement si $\overrightarrow{\lambda} \prec_{leximin} \overrightarrow{v(\mathbf{x})}$.*

Bien que cette contrainte n'existe pas en tant que telle dans la littérature, et donc *a fortiori* dans les systèmes de programmation par contraintes, on peut trouver dans le travail de [Frisch *et al.*, 2003] la description d'une contrainte très similaire, la contrainte **MultisetOrdering**, qui travaille sur les multi-ensembles . La sémantique de cette contrainte est la suivante : étant donnés deux multi-ensembles de variables $\mathscr{M}_1$ et $\mathscr{M}_2$, la contrainte **MultisetOrdering**$(\mathscr{M}_1, \mathscr{M}_2)$ porte sur toutes les variables de $\mathscr{M}_1$ et $\mathscr{M}_2$ et n'autorise que les valuations telles que $v(\mathscr{M}_1) \preceq_{multi} v(\mathscr{M}_2)$. L'ordre naturel $\preceq_{multi}$ sur les multi-ensembles est l'équivalent de l'ordre leximax sur les vecteurs, étendu au cas où les vecteurs peuvent être de taille différentes.

Moyennant quelques légères modifications, l'algorithme de filtrage introduit dans [Frisch *et al.*, 2003] peut être adapté à l'ordre leximin entre les vecteurs. Il assure l'arc-cohérence généralisée de cette contrainte en temps $O(n + d)$, où n est la longueur des vecteurs (ou des multi-ensembles), et d est la plus grande distance entre deux valeurs de tous les domaines des vecteurs. Dans le cas où cette valeur d est très grande, on peut bénéficier d'une variante de l'algorithme de filtrage, qui s'exécute en temps $O(n \log(n))$.

Proposition 5.1 *Si la fonction* **solve** *termine et est correcte, alors l'algorithme 1 termine et résout le problème [MAXLEXIMINCSP].*

Démonstration Si $sol(\mathscr{X}, \mathcal{D}, \mathscr{C}) = \varnothing$, alors le premier appel à **solve** renvoie «Inconsistant», et donc l'algorithme renvoie «Inconsistant». Dans le cas contraire, l'affectation $\widehat{v}$ est initialisée au premier passage de la boucle, et correspond à chaque itération à la dernière solution trouvée du réseau de contraintes additionné des contraintes **Leximin** des itérations précédentes. Donc l'algorithme renvoie une solution du réseau de contraintes augmenté des précédentes contraintes, donc *a fortiori* du réseau de contraintes initial. Notons v_r l'affectation retournée par l'algorithme, et supposons qu'il existe une instanciation v' solution du réseau de contraintes telle que $v'(\overrightarrow{\mathbf{u}}) \succ_{leximin} v_r(\overrightarrow{\mathbf{u}})$. v_r étant égale à l'instanciation optimale courante $\widehat{v}$ à la dernière itération, v' est donc une solution du réseau de contraintes de la ligne 5 à la dernière itération. Donc si **solve** est correcte, elle ne devrait pas retourner «Inconsistant». Or c'est le cas, car il s'agit de la dernière itération de l'algorithme. Il y a donc contradiction, ce qui prouve la proposition. ▲

5.4.3 Algorithmes itératifs

Contrairement à l'algorithme précédent, pour lequel l'optimisation était fondée directement sur le préordre leximin, les algorithmes suivants sont tous fondés sur une optimisation itérative, où, à chaque pas de l'algorithme, on essaie de maximiser la valeur d'une composante particulière de la version triée du vecteur objectif.

5.4.3.1 Brancher sur les sous-ensembles saturés

La solution algorithmique proposée dans le domaine des problèmes de satisfaction de contraintes flexibles [Dubois et Fortemps, 1999], ainsi que brièvement introduite dans [Ehrgott, 2000, page 145] est fondée sur une résolution de sous-problèmes maximin successifs. L'idée est de chercher, à chaque étape, tous les ensembles possibles de «pires» variables objectif, et de fixer explicitement leur valeur. Cette opération définit l'équivalent d'une «coupe-α forte» dans le domaine des problèmes de satisfaction de contraintes flexibles. Le terme «pires» fait référence à la notion de *sous-ensembles saturés de variables objectif* :

Définition 5.9 (Sous-ensemble saturé de variables objectif) *Soient $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ un réseau de contraintes et $\overrightarrow{\mathbf{u}}$ un vecteur de variables objectif. Soit $\widehat{m}$ la valeur maximale possible de la valeur la plus basse de $\overrightarrow{\mathbf{u}}$, ou en d'autres termes, $\widehat{m} = \max_{v \in sol(\mathscr{X}, \mathcal{D}, \mathscr{C})}\{\min_i\{v(\mathbf{u_i})\}\}$.*

Un sous-ensemble saturé de variables objectif *est un sous-ensemble $\mathscr{S}_{sat}$ de variables de $\overrightarrow{\mathbf{u}}$ tel qu'il existe $v \in sol(\mathscr{X}, \mathcal{D}, \mathscr{C})$ tel que $\forall \mathbf{x} \in \mathscr{S}_{sat}$, $v(\mathbf{x}) = \widehat{m}$ et $\forall \mathbf{y} \in \{\mathbf{u_i} \mid i \in [\![1, n]\!]\} \backslash \mathscr{S}_{sat}$, $v(\mathbf{y}) > \widehat{m}$.*

Clairement, les seuls sous-ensembles saturés qui peuvent conduire à une solution leximin-optimale sont ceux de cardinalité minimale. L'idée des algorithmes introduits dans [Dubois et Fortemps, 1999] pour le calcul de solutions leximin-optimales dans le contexte des problèmes de satisfaction de contraintes flexibles est fondée sur le calcul de ces sous-ensembles saturés de variables objectif de cardinalité minimale. Les algorithmes fonctionnent informellement de la manière suivante. Tout d'abord, on calcule la valeur maximin $\widehat{m}$ et les sous-ensembles saturés de variables objectif de cardinalité minimale. Ensuite, pour chaque sous-ensemble $\mathscr{S}_{sat}$ on enlève chaque variable de $\mathscr{S}_{sat}$ du vecteur objectif, et on fixe sa valeur à $\widehat{m}$. Puis on effectue la même opération pour chaque nouveau vecteur objectif obtenu, jusqu'à ce que l'on n'ait plus de variable.

Il peut y avoir, dans le cas général, plusieurs sous-ensembles saturés de cardinalité minimale à chaque pas de l'algorithme. L'algorithme peut donc être vu comme une procédure de branchement qui choisit à chaque nœud sur quel sous-ensemble saturé il va poursuivre l'exploration. La traduction dans le cadre de la programmation par contraintes de l'algorithme de recherche en profondeur d'abord introduit dans [Dubois et Fortemps, 1999] est présenté dans l'algorithme 2. Il est fondé sur la fonction `Explore`, qui est appelée de manière récursive afin d'explorer l'arbre de recherche : à chaque nœud, cette fonction calcule la valeur maximin (à l'aide de la fonction `FindMaximin`, qui effectue un appel à la fonction **maximize**), puis calcule les sous-ensembles saturés de cardinalité minimale (à l'aide de la fonction `FindSaturatedSubsets` qui effectue plusieurs appels à la fonction **solve**), puis explore successivement les sous-arbres induits par ces sous-ensembles saturés. À la toute fin de l'algorithme, une comparaison leximin est effectuée, car certaines branches de l'arbre de recherche peuvent conduire à des solutions sous-optimales.

Exemple 5.1.b L'arbre de recherche développé par l'algorithme 2 pour l'exemple 5.1 est tracé sur la figure 5.2. Sur le côté gauche de la figure, on peut voir les sous-ensembles saturés de variables objectif, et sur le côté droit les apparaissent les solutions restantes pour chaque nœud

Algorithme 2 — Résolution du problème [MAXLEXIMINCSP] en branchant sur des sous-ensembles saturés (version DFS).

entrée : Un réseau de contraintes $(\mathscr{X}, \mathcal{D}, \mathscr{C})$; un vecteur objectif $(\mathbf{u_1}, \ldots, \mathbf{u_n}) \in \mathscr{X}^n$.
sortie : Une solution au problème [MAXLEXIMINCSP].

$sol \leftarrow$ `Explore` $((\mathscr{X}, \mathcal{D}, \mathscr{C}), (\mathbf{u_1}, \ldots, \mathbf{u_n}))$;
si $sol = \varnothing$ **alors retourner** *«Inconsistant»*;
retourner LeximinOptimal(sol); `/* Comparaison leximin de toutes les solutions trouvées. */`

Fonction `Explore`$((\mathscr{X}, \mathcal{D}, \mathscr{C}), (\mathbf{u_1}, \ldots, \mathbf{u_k}))$

entrée : Un réseau de contraintes $(\mathscr{X}, \mathcal{D}, \mathscr{C})$; un vecteur objectif $(\mathbf{u_1}, \ldots, \mathbf{u_k}) \in \mathscr{X}^k$.
sortie : Un ensemble de solutions leximin-optimales potentielles.

si $\overrightarrow{\mathbf{u}} = \varnothing$ **alors retourner solve**$(\mathscr{X}, \mathcal{D}, \mathscr{C})$;
$\widehat{m} \leftarrow$ `FindMaximin`$((\mathscr{X}, \mathcal{D}, \mathscr{C}), (\mathbf{u_1}, \ldots, \mathbf{u_k}))$;
$sat \leftarrow$ `FindMinimalSaturatedSubsets`$((\mathscr{X}, \mathcal{D}, \mathscr{C}), (\mathbf{u_1}, \ldots, \mathbf{u_k}), \widehat{m})$;
$sol \leftarrow \varnothing$;
pour tous les $\mathcal{S} \in sat$ **faire**
  pour tous les $\mathbf{u_i} \in \mathcal{S}$ **faire** $\mathbf{u_i} \leftarrow \widehat{m}$;
  pour tous les $\mathbf{u_i} \notin \mathcal{S}$ **faire** $\underline{\mathbf{u_i}} \leftarrow \widehat{m} + 1$;
  $sol \leftarrow sol \cup$ `Explore`$((\mathscr{X}, \mathcal{D}, \overline{\mathscr{C})}, (\mathbf{u_1}, \ldots, \mathbf{u_k}) \setminus \mathcal{S})$
retourner sol;

de l'arbre de recherche. L'ensemble *sol* retourné par l'appel à `Explore` dans l'algorithme 2 est $\{(5,3,6),(7,3,5),(5,5,3)\}$.

La principale difficulté de cet algorithme est liée au calcul de sous-ensembles saturés, et, puisque dans le cas général il peut y en avoir plusieurs, de brancher sur ces sous-ensembles. Cependant, dans un certain nombre de cas connus, il existe à chaque pas de l'algorithme un sous-ensemble saturé inclus dans tous les autres, et donc un seul et unique sous-ensemble saturé de cardinalité minimale. En d'autres termes, la fonction `FindMinimalSaturatedSubsets` ne retourne qu'un seul sous-ensemble saturé à chaque pas. Dans ces cas-là, l'algorithme 2 ne produit aucun branchement, et il suffit de choisir à chaque pas l'unique sous-ensemble saturé de cardinalité minimale. Cela arrive typiquement dans les problèmes linéaires continus pour lesquels l'ensemble des alternatives est convexe [Ehrgott, 2000; Ogryczak, 1997; Luss, 1999; Potters et Tijs, 1992], ce qui explique le succès et l'efficacité de cet algorithme dans ce contexte. Un exemple d'application de l'algorithme sur un problème linéaire continu à 5 variables objectifs est illustré dans la figure 5.3.

Fonction `FindMinimalSaturatedSubsets`$((\mathscr{X}, \mathcal{D}, \mathscr{C}), (\mathbf{u_1}, \ldots, \mathbf{u_k}), \widehat{m})$

entrée : Un réseau de contraintes $(\mathscr{X}, \mathcal{D}, \mathscr{C})$; un vecteur objectif $(\mathbf{u_1}, \ldots, \mathbf{u_k}) \in \mathscr{X}^k$; un entier $\widehat{m}$.
sortie : L'ensemble de sous-ensembles saturés de cardinalité minimale du vecteur objectif pour $\widehat{m}$.

$sat \leftarrow \varnothing$; $i \leftarrow 1$;
tant que $i \leq k$ **et** $sat = \varnothing$ **faire**
  pour tous les $\mathcal{S} \subset \{\overrightarrow{\mathbf{u}}\}$ **tels que** $|\mathcal{S}| = i$ **faire**
    si solve$(\mathscr{X}, \mathcal{D}, \mathscr{C} \cup \bigcup_{\mathbf{u_j} \in \mathcal{S}} \{\mathbf{u_j} = \widehat{m}\} \cup \bigcup_{\mathbf{u_j} \notin \mathcal{S}} \{\mathbf{u_j} > \widehat{m}\}) \neq$ *«Inconsistant»* **alors**
      $sat \leftarrow sat \cup \mathcal{S}$;
  $i \leftarrow i + 1$;
retourner sat;

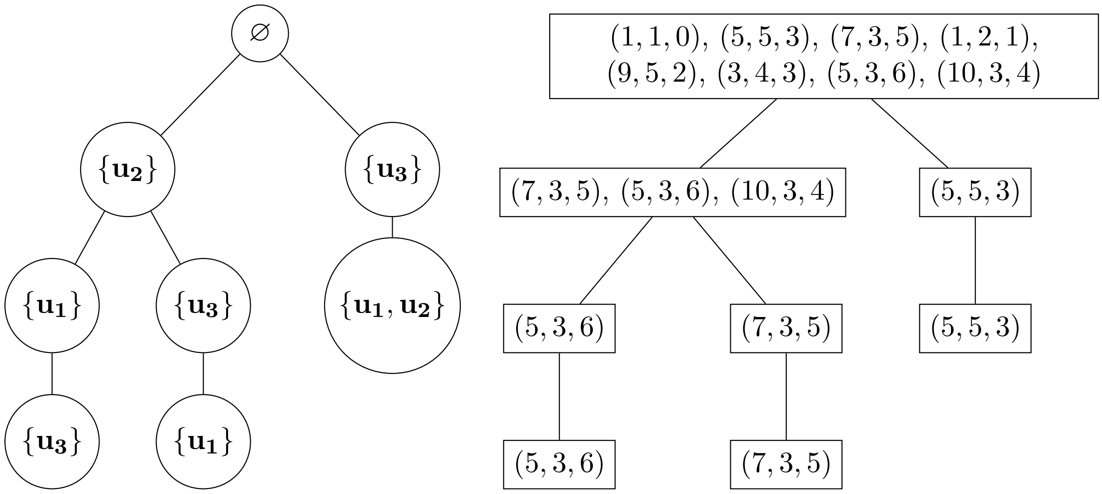
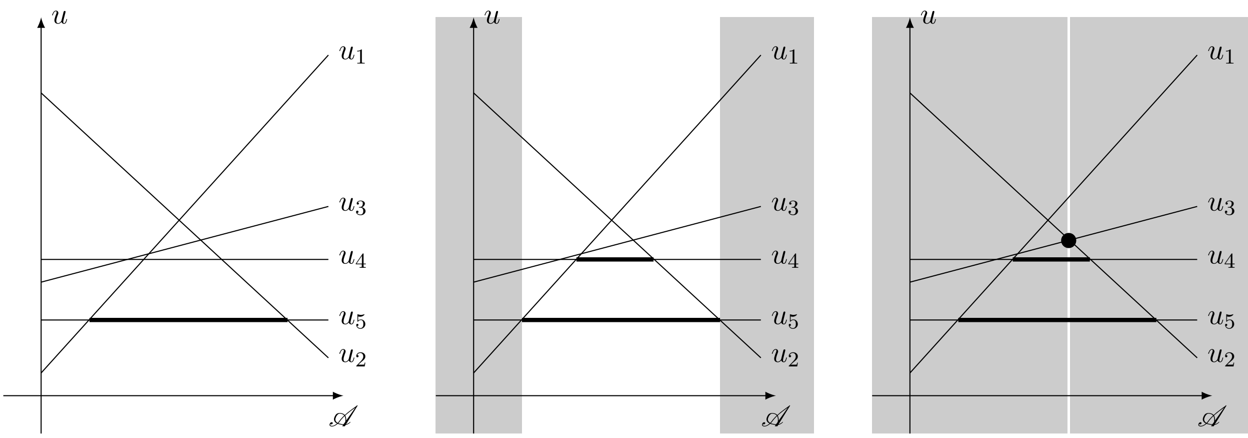


Figure 5.2 — L'arbre de recherche développé par l'algorithme 2 pour l'exemple 5.1.



L'axe des abscisses représente l'ensemble des alternatives admissibles, l'axe des ordonnées représente les valeurs des variables objectif. Les 5 droites sont les représentations graphiques des valeurs des 5 variables objectifs sur l'ensemble des alternatives.
Les traits gras représentent le lieu des solutions maximin à chaque pas. Les sous-ensembles saturés correspondant sont respectivement $\{\mathbf{u_5}\}$, $\{\mathbf{u_4}\}$ et $\{\mathbf{u_2}, \mathbf{u_3}\}$. Les zones grisées correspondent aux restrictions de domaines induites en fixant la valeur des variables des sous-ensembles saturés à la valeur maximin.

Figure 5.3 — Illustration de l'algorithme 2 sur un problème linéaire continu.

5.4.3.2 Trier pour régner

Cependant, dans un contexte de problèmes discrets tels que les [MAXLEXIMINCSP], il peut y avoir plusieurs sous-ensembles saturés de cardinalité minimale à chaque pas de l'algorithme, et donc leur calcul peut s'avérer très coûteux, ce qui au final rend ces algorithmes inutilisables en pratique. L'une des solutions pour résoudre ce problème est d'introduire de nouvelles variables pour remplacer les variables objectif en faisant en sorte que (1) l'introduction de ces variables ne modifie par l'ensemble des solutions leximin-optimales, et que (2) cela garantisse l'unicité du sous-ensemble saturé de cardinalité minimale.

Une manière intuitive de procéder est d'introduire de manière plus ou moins explicite la version triée du vecteur objectif. Les solutions leximin-optimales vis-à-vis du vecteur objectif trié sont très clairement les mêmes que les solutions leximin-optimales relatives au vecteur objectif non trié. De plus, les seuls sous-ensembles saturés sont constitués des k premières composantes du vecteur objectif trié, et donc les solutions leximin-optimales peuvent être calculées par des maximisations successives des premières composantes de ce vecteur trié.

Le cadre de la programmation par contraintes nous permet d'introduire de manière naturelle la version triée $\overrightarrow{\mathbf{y}}$ du vecteur objectif $\overrightarrow{\mathbf{u}}$ à l'aide d'une contrainte $\mathbf{Sort}(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{y}})$, définie comme suit :

Définition 5.10 (Contrainte Sort) *Soient $\overrightarrow{\mathbf{x}}$ et $\overrightarrow{\mathbf{x}}'$ deux vecteurs de variables de même longueur, et v une instanciation. La contrainte $\mathbf{Sort}(\overrightarrow{\mathbf{x}}, \overrightarrow{\mathbf{x}}')$ porte sur $\overrightarrow{\mathbf{x}} \cup \overrightarrow{\mathbf{x}}'$, et est satisfaite par v si et seulement si $v(\overrightarrow{\mathbf{x}}')$ est la version triée de $v(\overrightarrow{\mathbf{x}})$ dans l'ordre croissant.*

Cette contrainte a été étudiée en particulier dans deux articles, qui introduisent tous deux un algorithme de filtrage pour assurer la cohérence de borne sur cette contrainte. Le premier algorithme vient de [Bleuzen-Guernalec et Colmerauer, 1997] et s'exécute en temps $O(n \log n)$ (n étant la taille de $\overrightarrow{\mathbf{x}}$). Quelques années plus tard, [Mehlhorn et Thiel, 2000] développe un algorithme qui nécessite un temps d'exécution de $O(n)$ plus le temps requis par le tri des bornes de l'intervalle de $\overrightarrow{\mathbf{x}}$, ce qui peut s'avérer asymptotiquement plus rapide que $O(n \log n)$.

Notre méthode de calcul d'une solution leximin-optimale fondée sur la contrainte **Sort** (présentée dans l'algorithme 5) fonctionne de informellement de la manière suivante : ayant introduit la version triée $\overrightarrow{\mathbf{y}}$ du vecteur objectif $\overrightarrow{\mathbf{u}}$, elle maximise successivement les composantes de ce vecteur, sachant que la solution leximin-optimale est la solution qui maximise $\mathbf{y_1}$, et, étant donnée cette valeur maximale, maximise $\mathbf{y_2}$, *et caetera* jusqu'à $\mathbf{y_n}$.

Exemple 5.1.c Revenons sur notre exemple. Au début de l'algorithme, 3 nouvelles variables $(\mathbf{y_1}, \mathbf{y_2}, \mathbf{y_3})$ sont introduites, afin de représenter la version triée du vecteur objectif. Les instanciations admissibles pour $(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{y}})$ sont : $((1,1,0),(0,1,1))$, $((5,5,3),(3,5,5))$, $((7,3,5),(3,5,7))$, $((1,2,1),(1,1,2))$, $((9,5,2),(2,5,9))$, $((3,4,3),(3,3,4))$, $((5,3,6),(3,5,6))$ et $((10,3,4),(3,4,10))$.

- ▷ Pendant le premier pas de l'algorithme, la variable $\mathbf{y_1}$ est maximisée (sa valeur maximale est 3) et ensuite est fixée à sa valeur optimale 3. Les instanciations admissibles restantes sont donc : $((5,5,3),(\mathbf{3},5,5))$, $((7,3,5),(\mathbf{3},5,7))$, $((3,4,3),(\mathbf{3},3,4))$, $((5,3,6),(\mathbf{3},5,6))$ et $((10,3,4),(\mathbf{3},4,10))$.
- ▷ Pendant le second pas de l'algorithme, la variable $\mathbf{y_2}$ est maximisée (sa valeur maximale est 5), et ensuite est fixée à sa valeur optimale 5. Les instanciations admissibles restantes sont donc : $((5,5,3),(3,\mathbf{5},5))$, $((7,3,5),(3,\mathbf{5},7))$ et $((5,3,6),(3,\mathbf{5},6))$.
- ▷ Pendant le troisième et dernier pas de l'algorithme, $\mathbf{y_3}$ est maximisée (sa valeur maximale est 7) L'unique solution leximin-optimale est : $((7,3,5),(3,5,\mathbf{7}))$.

Proposition 5.2 *Si les deux fonctions* **maximize** *et* **solve** *sont correctes et terminent, alors l'algorithme 5 termine et renvoie une solution au problème [MAXLEXIMINCSP].*

Algorithme 5 — Résolution du problème [MAXLEXIMINCSP] en utilisant une contrainte de tri (Trier pour régner).

entrée : Un réseau de contraintes $(\mathscr{X}, \mathcal{D}, \mathscr{C})$; un vecteur objectif $(\mathbf{u_1}, \ldots, \mathbf{u_n}) \in \mathscr{X}^n$.
sortie : Une solution au problème [MAXLEXIMINCSP].

1 **si** **solve**$(\mathscr{X}, \mathscr{D}, \mathscr{C}) =$«*Inconsistant*» **retourner** «*Inconsistant*»;
2 $\mathscr{X}' \leftarrow \mathscr{X} \cup \{\mathbf{y_1}, \ldots, \mathbf{y_n}\}$;
3 $\mathcal{D}' \leftarrow \langle \mathscr{D}, (\mathbf{y_1} : \mathscr{D}_{\mathbf{y_1}}, \ldots, \mathbf{y_n} : \mathscr{D}_{\mathbf{y_n}}) \rangle$ avec $\mathscr{D}_{\mathbf{y_i}} = [\![\min_j(\underline{\mathbf{u_j}}), \max_j(\overline{\mathbf{u_j}})]\!]$;
4 $\mathscr{C}' \leftarrow \mathscr{C} \cup \{\mathbf{Sort}(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{y}})\}$;
5 **pour** $i \leftarrow 1$ **à** n **faire**
6 $\quad \widehat{v}_{(i)} \leftarrow$ **maximize**$(\mathscr{X}', \mathscr{D}', \mathscr{C}', \mathbf{y_i})$;
7 $\quad \mathbf{y_i} \leftarrow \widehat{v}_{(i)}(\mathbf{y_i})$;
8 **retourner** $\widehat{v}_{(n)\downarrow\mathscr{X}}$;

Démonstration Si $sol(\mathscr{X}, \mathcal{D}, \mathscr{C}) = \varnothing$ et si la fonction **solve** est correcte, alors l'algorithme 5 retourne «Inconsistant» de manière évidente. Nous supposerons dans la suite de la preuve que nous ne sommes pas dans ce cas-là, et nous emploierons les notations suivantes : $\mathscr{S}_i$ et $\mathscr{S}'_i$ seront respectivement les ensembles de solutions du réseau de contrainte $(\mathscr{X}', \mathcal{D}', \mathscr{C}')$ au début et à la fin de l'itération i.

Nous avons de manière évidente $\mathscr{S}_{i+1} = \mathscr{S}'_i$ pour tout $i \in [\![1, n-1]\!]$, ce qui prouve que si $\mathscr{S}_i \neq \varnothing$, alors l'appel à **maximize** à la ligne 6 ne renvoie pas «Inconsistant», et $\mathscr{S}_{i+1} \neq \varnothing$. Ainsi, $\widehat{v}_{(n)}$ est bien défini et $(\widehat{v}_{(n)})_{\downarrow\mathscr{X}}$ est clairement une solution de $(\mathscr{X}, \mathcal{D}, \mathscr{C})$.

Nous notons $\widehat{v} = \widehat{v}_{(n)}$ l'instanciation calculée par le dernier appel à **maximize** dans l'algorithme 5. Supposons qu'il y ait une instanciation $v \in sol(\mathscr{X}, \mathcal{D}, \mathscr{C})$ telle que $\widehat{v}(\overrightarrow{\mathbf{u}}) \prec_{leximin} v(\overrightarrow{\mathbf{u}})$. Nous définissons alors v^+ comme étant l'extension de v qui instancie chaque $\mathbf{y_i}$ à $v(\overrightarrow{\mathbf{u}})^{\uparrow}_i$. En raison de la contrainte **Sort**, $\widehat{v}(\overrightarrow{\mathbf{y}})$ et $v^+(\overrightarrow{\mathbf{y}})$ sont les versions triées respectives de $\widehat{v}(\overrightarrow{\mathbf{u}})$ et $v^+(\overrightarrow{\mathbf{u}})$. D'après la définition 1.32 du préordre leximin (au changement d'indice près), il existe un indice $i \in [\![0, n-1]\!]$ tel que $\forall j \in [\![1, i]\!]$, $\widehat{v}(\mathbf{y_j}) = v^+(\mathbf{y_j})$ et $\widehat{v}(\mathbf{y_{i+1}}) < v^+(\mathbf{y_{i+1}})$. À cause de la ligne 7, on a $\widehat{v}(\mathbf{y_{i+1}}) = \widehat{v}_{(n)}(\mathbf{y_{i+1}}) = \widehat{v}_{(i+1)}(\mathbf{y_{i+1}})$. Donc v^+ est une solution appartenant à l'ensemble $max(\mathscr{X}', \mathscr{D}', \mathscr{C}', \mathbf{y_{i+1}})$ dont la valeur de la variable objectif $v^+_{(i+1)}(\mathbf{y_{i+1}})$ est strictement plus grande que $\widehat{v}_{(i+1)}(\mathbf{y_{i+1}})$, ce qui contredit l'hypothèse de correction de **maximize**. Il n'existe donc pas de solution v de $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ telle que $\widehat{v}(\overrightarrow{\mathbf{u}}) \prec_{leximin} v(\overrightarrow{\mathbf{u}})$: donc $\widehat{v}(\overrightarrow{\mathbf{u}})$ est une solution leximin-optimale.
▲

5.4.3.3 Un nouvel algorithme utilisant une méta-contrainte de cardinalité

Avant de présenter ce nouvel algorithme, nous introduisons la notation suivante : étant donnés un vecteur de nombres entiers $\overrightarrow{x}$ et un entier α, $\sum_i(\alpha \leq x_i)$ désignera la cardinalité de l'ensemble $\{i \mid \alpha \leq x_i\}$. Cette notation est inspirée par la notion de réification dans un langage de programmation par contraintes tel qu'OPL [van Hentenryck, 1999], où $(\alpha \leq x_i)$ représente une valeur booléenne valant 1 si l'inégalité est satisfaite et 0 sinon.

L'algorithme précédent introduisait de manière explicite la version triée du vecteur objectif, nécessitant donc une propagation de la contrainte **Sort** portant sur l'intégralité de ce vecteur trié. Il est cependant possible, grâce à une petite astuce, d'accéder directement à la $i^{\text{ème}}$ composante du vecteur objectif trié sans recourir de manière explicite à l'intégralité du vecteur trié. Le nouvel algorithme que nous allons présenter exploite cette remarque, qui s'appuie sur la proposition (évidente) suivante :

Proposition 5.3 *Soit $\overrightarrow{x}$ un vecteur de nombres de taille n. Nous avons :*

$$x_i^{\uparrow} = \max\{\alpha \mid \sum_i (\alpha \leq x_i) \geq n - i + 1\}$$

En d'autres termes, le $i^{\text{ème}}$ minimum d'un vecteur de nombres de taille n est le nombre maximal α tel qu'au moins $n - i + 1$ composantes du vecteur sont supérieures ou égales à α.

L'algorithme fondé sur la méta-contrainte AtLeast Cette nouvelle approche du problème, présentée dans l'algorithme 6, fonctionne de manière relativement similaire à l'algorithme précédent, mais comme nous l'avons indiqué, il ne nécessite pas l'introduction de la version triée du vecteur objectif dans son intégralité. De manière informelle, il fonctionne comme suit :

- ▷ il calcule dans un premier temps la valeur maximale $\widehat{y_1}$ de $\mathbf{y_1}$ telle qu'il existe une solution v avec $\sum_i (\widehat{y_1} \leq v(\mathbf{u_i})) = n$ (ou en d'autres termes $\forall i,\ \widehat{y_1} \leq v(\mathbf{u_i})$) ;
- ▷ ensuite il fixe $\mathbf{y_1}$ à $\widehat{y_1}$ et calcule la valeur maximale $\widehat{y_2}$ de $\mathbf{y_2}$ telle qu'il existe une solution v avec $\sum_i (\widehat{y_2} \leq v(\mathbf{u_i})) \geq n - 1$;
- ▷ *et caetera* jusqu'à ce que, ayant fixé $\mathbf{y_{n-1}}$ à $\widehat{y_{n-1}}$, il calcule la valeur maximale $\widehat{y_n}$ de $\mathbf{y_n}$ telle qu'il existe une solution v avec $\sum_i (\widehat{y_n} \leq v(\mathbf{u_i})) \geq 1$.

Afin d'assurer la contrainte sur les $\mathbf{u_i}$, nous utilisons la méta-contrainte **AtLeast**, dérivée d'un *combinateur de cardinalité* introduit dans [van Hentenryck *et al.*, 1992], et existant dans la plupart des systèmes de programmation par contraintes :

Définition 5.11 (Méta-contrainte AtLeast) *Soient Γ un ensemble de p contraintes, et $k \in [\![1, p]\!]$ un entier. La méta-contrainte* $\mathbf{AtLeast}(\Gamma, k)$ *porte sur l'union des scopes de Γ et est satisfaite par v si et seulement si au moins k contraintes de Γ sont satisfaites v.*

Cette approche, présentée dans l'algorithme 6, est illustrée dans l'exemple suivant :

Exemple 5.1.d Nous appliquons l'algorithme 6 sur l'exemple 5.1 :

- ▷ Lors du premier pas de l'algorithme, une variable $\mathbf{y_1}$ est introduite, et l'on impose à toutes les variables objectif de prendre une valeur supérieure à celle de $\mathbf{y_1}$. Cela donne les solutions suivantes pour $(\overrightarrow{\mathbf{u}}, \mathbf{y_1})$: $((1,1,0),0)$, $((5,5,3),[\![0,\mathbf{3}]\!])$, $((7,3,5),[\![0,\mathbf{3}]\!])$, $((1,2,1),[\![0,1]\!])$, $((9,5,2),[\![0,2]\!])$, $((3,4,3),[\![0,\mathbf{3}]\!])$, $((5,3,6),[\![0,\mathbf{3}]\!])$ et $((10,3,4),[\![0,\mathbf{3}]\!])$. $\mathbf{y_1}$ est fixée à sa valeur maximale 3 (apparaissant en gras ci-avant), ce qui a pour effet de restreindre l'ensemble des instanciations admissibles à $\{((5,5,3),3),((7,3,5),3),((3,4,3),3),((5,3,6),3),((10,3,4),3)\}$.
- ▷ Lors du deuxième pas de l'algorithme, une variable $\mathbf{y_2}$ est introduite, et l'on impose qu'il y ait au moins deux variables objectif qui aient une valeur supérieure à celle de $\mathbf{y_2}$. Cela donne les solutions suivantes pour $(\overrightarrow{\mathbf{u}}, \mathbf{y_2})$: $((5,5,3),[\![0,\mathbf{5}]\!])$, $((7,3,5),[\![0,\mathbf{5}]\!])$, $((3,4,3),[\![0,4]\!])$, $((5,3,6),[\![0,\mathbf{5}]\!])$ et $((10,3,4),[\![0,4]\!])$. $\mathbf{y_2}$ est fixée à sa valeur maximale 5 (apparaissant en gras ci-avant), ce qui a pour effet de restreindre l'ensemble des instanciations admissibles à $\{((5,5,3),5),((7,3,5),5),((5,3,6),5)\}$.
- ▷ Lors du troisième pas de l'algorithme, une variable $\mathbf{y_3}$ est introduite, et l'on impose qu'il y ait au moins une variable objectif qui ait une valeur supérieure à celle de $\mathbf{y_3}$. Cela donne les solutions suivantes pour $(\overrightarrow{\mathbf{u}}, \mathbf{y_3})$: $((5,5,3),[\![0,5]\!])$, $((7,3,5),[\![0,\mathbf{7}]\!])$ et $((5,3,6),[\![0,6]\!])$. La valeur maximale de $\mathbf{y_3}$ est 7 (écrite en gras ci-avant), ce qui conduit à l'unique solution leximin-optimale $(7,3,5)$.

Proposition 5.4 *Si les fonctions* **maximize** *et* **solve** *sont toutes deux correctes et terminent, alors l'algorithme 6 termine et retourne une solution du problème [MAXLEXIMINCSP].*

Algorithme 6 — Résolution du problème [MAXLEXIMINCSP] en utilisant une méta-contrainte de cardinalité.

entrée : Un réseau de contraintes $(\mathscr{X}, \mathcal{D}, \mathscr{C})$; un vecteur objectif $(\mathbf{u_1}, \dots, \mathbf{u_n}) \in \mathscr{X}^n$.
sortie : Une solution au problème [MAXLEXIMINCSP].

1 **si solve**$(\mathscr{X}, \mathcal{D}, \mathscr{C}) =$ «*Inconsistant*» **retourner** «*Inconsistant*»;
2 $(\mathscr{X}_0, \mathcal{D}'_0, \mathscr{C}_0) \leftarrow (\mathscr{X}, \mathcal{D}, \mathscr{C})$;
3 **pour** $i \leftarrow 1$ **à** n **faire**
4 $\quad \mathscr{X}_i \leftarrow \mathscr{X}_{i-1} \cup \{\mathbf{y_i}\}$;
5 $\quad \mathcal{D}_i \leftarrow \langle \mathcal{D}'_{i-1}, (\mathbf{y_i} : \mathscr{D}_{\mathbf{y_i}}) \rangle$ avec $\mathscr{D}_{\mathbf{y_i}} = [\![\min_j(\underline{\mathbf{u_j}}), \max_j(\overline{\mathbf{u_j}})]\!]$;
6 $\quad \mathscr{C}_i \leftarrow \mathscr{C}_{i-1} \cup \{\mathbf{AtLeast}(\{\mathbf{y_i} \leq \mathbf{u_1}, \dots, \mathbf{y_i} \leq \mathbf{u_n}\}, n - i + 1)\}$;
7 $\quad \widehat{v}_{(i)} \leftarrow$ **maximize**$(\mathscr{X}_i, \mathcal{D}_i, \mathscr{C}_i, \mathbf{y_i})$;
8 $\quad \mathcal{D}'_i \leftarrow \mathcal{D}_i$ avec $\mathbf{y_i} \leftarrow \widehat{v}_{(i)}(\mathbf{y_i})$;

9 **retourner** $\widehat{v}_{(n)\downarrow\mathscr{X}}$;

Dans les preuves suivantes, nous écrirons sol_i et sol'_i pour désigner respectivement $sol(\mathscr{X}_i, \mathcal{D}_i, \mathscr{C}_i)$ et $sol(\mathscr{X}_i, \mathcal{D}'_i, \mathscr{C}_i)$. Nous noterons de même $(sol_i)_{\downarrow\mathscr{X}_j}$ et $(sol'_i)_{\downarrow\mathscr{X}_j}$ pour désigner les mêmes ensembles de solutions projetés sur $\mathscr{X}_j$ (avec $j < i$). Nous pouvons remarquer dès à présent que $sol_0 = sol(\mathscr{X}, \mathcal{D}, \mathscr{C})$, et que $\forall i$, $sol'_i \subseteq sol_i$ (à cause de la ligne 8 qui restreint le domaine de $\mathbf{y_i}$).

Lemme 18 *Si $sol_0 \neq \varnothing$, alors $\widehat{v}_{(n)}$ est bien défini et n'est pas égal à «Inconsistant».*

Démonstration Soit $i \in [\![1, n]\!]$. Supposons que $sol'_{i-1} \neq \varnothing$, et soit $v_{(i)} \in sol'_{i-1}$. Alors l'extension de $v_{(i)}$ qui instancie $\mathbf{y_i}$ à $\min_j(\underline{\mathbf{u_j}})$ est une solution de $(\mathscr{X}_i, \mathcal{D}_i, \mathscr{C}_i)$ (puisqu'une seule contrainte a été ajoutée entre $\mathscr{C}_{i-1}$ et $\mathscr{C}_i$ et qu'elle est satisfaite de manière évidente par cette dernière instanciation). En conséquence, $sol_i \neq \varnothing$, et, si **maximize** est correcte, $\widehat{v}_{(i)} \neq$ «Inconsistant» et $\widehat{v}_{(i)} \in sol'_i$. Ainsi, $sol'_i \neq \varnothing$, ce qui prouve le lemme 18 par récurrence. ▲

Lemme 19 *Si $sol_0 \neq \varnothing$, alors $(\widehat{v}_{(n)})_{\downarrow\mathscr{X}_i} \in sol_i, \forall i \in [\![0, n]\!]$.*

Démonstration On a $sol'_i \subseteq sol_i$, et $(sol_{i+1})_{\downarrow\mathscr{X}_i} \subseteq sol'_i$ (puisque de $(\mathscr{X}_i, \mathscr{D}'_i, \mathscr{C}_i)$ à $(\mathscr{X}_{i+1}, \mathcal{D}_{i+1}, \mathscr{C}_{i+1})$ on a simplement ajouté une contrainte). Plus généralement, on a $(sol'_i)_{\downarrow\mathscr{X}_j} \subseteq (sol_i)_{\downarrow\mathscr{X}_j}$, et $(sol_{i+1})_{\downarrow\mathscr{X}_j} \subseteq (sol'_i)_{\downarrow\mathscr{X}_j}$, pour peu que $j \leq i$. Ainsi, $(\widehat{v}_{(n)})_{\downarrow\mathscr{X}_i} \in (sol'_n)_{\downarrow\mathscr{X}_i} \subseteq (sol_n)_{\downarrow\mathscr{X}_i} \subseteq \cdots \subseteq (sol_{i+1})_{\downarrow\mathscr{X}_i} \subseteq sol'_i \subseteq sol_i$. ▲

Lemme 20 *Si $sol_0 \neq \varnothing$, $\widehat{v}_{(n)}(\overrightarrow{\mathbf{y}})$ est égal à $\widehat{v}_{(n)}(\overrightarrow{\mathbf{u}})^{\uparrow}$.*

Démonstration Pour tout $i \in [\![1, n]\!]$, $(\widehat{v}_{(n)})_{\downarrow\mathscr{X}_i}$ est une solution de sol_i d'après le lemme 19. D'après la proposition 5.3, l'instanciation $(\widehat{v}_{(n)})_{\downarrow\mathscr{X}_i}$ dans laquelle on a remplacé la valeur de $\mathbf{y_i}$ par $\widehat{v}_{(n)}(\overrightarrow{\mathbf{u}})^{\uparrow}_i]$ satisfait la contrainte de cardinalité à l'itération i, et est donc une solution de sol_i. Par définition de la fonction **maximize**, on a donc $\widehat{v}_{(i)}(\mathbf{y_i}) \geq \widehat{v}_{(n)}(\overrightarrow{\mathbf{u}})^{\uparrow}_i$. Puisque $\widehat{v}_{(i)}(\mathbf{y_i}) = \widehat{v}_{(n)}(\mathbf{y_i})$, on a $\widehat{v}_{(n)}(\mathbf{y_i}) \geq \widehat{v}_{(n)}(\overrightarrow{\mathbf{u}})^{\uparrow}_i$.

Puisque $\widehat{v}_{(n)}$ est une solution de sol_n, au moins $n-i+1$ nombres parmi ceux du vecteur $\widehat{v}_{(n)}(\overrightarrow{\mathbf{u}})$ sont supérieurs ou égaux à $\widehat{v}_{(n)}(\mathbf{y_i})$. Donc, les $n-i+1$ composantes les plus grandes de $\widehat{v}_{(n)}(\overrightarrow{\mathbf{u}})$ au moins doivent être supérieures ou égales à $\widehat{v}_{(n)}(\mathbf{y_i})$. Ces composantes incluent $\widehat{v}_{(n)}(\overrightarrow{\mathbf{u}})^{\uparrow}_i$, ce qui montre que $\widehat{v}_{(n)}(\mathbf{y_i}) \leq \widehat{v}_{(n)}(\overrightarrow{\mathbf{u}})^{\uparrow}_i$, ce qui prouve enfin le lemme. ▲

Nous pouvons maintenant rassembler tous les résultats précédents et prouver la proposition 5.4.

Démonstration (proposition 5.4) Si $sol(\mathscr{X}, \mathcal{D}, \mathscr{C}) = \varnothing$, et si **solve** est correcte, alors l'algorithme 6 retourne de manière évidente «Inconsistant». Sinon, d'après le lemme 18, cet algorithme retourne une instanciation $(\widehat{v}_{(n)})_{\downarrow \mathscr{X}}$ qui est, d'après le lemme 19, une solution de $(\mathscr{X}_0, \mathcal{D}_0, \mathscr{C}_0) = (\mathscr{X}, \mathcal{D}, \mathscr{C})$.

Supposons qu'il existe une instanciation $v \in sol(\mathscr{X}, \mathcal{D}, \mathscr{C})$ telle que $\widehat{v}_{(n)}(\overrightarrow{\mathbf{u}}) \prec_{leximin} v(\overrightarrow{\mathbf{u}})$. Alors, d'après la définition 1.32 du préordre leximin (au changement d'indice près), $\exists i \in [\![1, n]\!]$ tel que $\forall j < i$, $v(\overrightarrow{\mathbf{u}})^{\uparrow}_j = \widehat{v}_{(n)}(\overrightarrow{\mathbf{u}})^{\uparrow}_j$ et $\widehat{v}_{(n)}(\overrightarrow{\mathbf{u}})^{\uparrow}_i < v(\overrightarrow{\mathbf{u}})^{\uparrow}_i$. Soit $v^+_{(i)}$ l'extension de v qui instancie $\mathbf{y_1}, \dots \mathbf{y_{i-1}}$ aux valeurs respectives $\widehat{v}_{(n)}(\mathbf{y_1}), \dots \widehat{v}_{(n)}(\mathbf{y_{i-1}})$ et $\mathbf{y_i}$ à $v(\overrightarrow{\mathbf{u}})^{\uparrow}_i$. D'après le lemme 20, $\forall j$, $\widehat{v}_{(n)}(\mathbf{y_j}) = \widehat{v}_{(n)}(\overrightarrow{\mathbf{u}})^{\uparrow}_j$. En réunissant toutes les égalités précédentes, nous obtenons $\forall j < i$ $v^+_{(i)}(\mathbf{y_j}) = \widehat{v}_{(n)}(\mathbf{y_j}) = v(\overrightarrow{\mathbf{u}})^{\uparrow}_j = (v^+_{(i)}(\overrightarrow{\mathbf{u}}))^{\uparrow}_j$. On a aussi $v^+_{(i)}(\mathbf{y_i}) = v(\overrightarrow{\mathbf{u}})^{\uparrow}_i = (v^+_{(i)}(\overrightarrow{\mathbf{u}}))^{\uparrow}_i$. D'après la proposition 5.3, $\forall j \leq i$ au moins $n - j + 1$ composantes de $(v^+_{(i)}(\overrightarrow{\mathbf{u}}))$ sont supérieures ou égales à $v^+_{(i)}(\mathbf{y_j})$, ce qui prouve que $v^+_{(i)}$ satisfait toutes les contraintes de cardinalité à l'itération i. Puisque cette instanciation satisfait aussi toutes les contraintes de $\mathscr{C}$ et instancie chaque variable de $\mathscr{X}_i$ à l'une de ses valeurs possibles, il s'agit d'une solution de sol_i, et $v^+_{(i)}(\mathbf{y_i}) = v(\overrightarrow{\mathbf{u}})^{\uparrow}_i > \widehat{v}_{(n)}(\overrightarrow{\mathbf{u}})^{\uparrow}_i = \widehat{v}_{(i)}(\mathbf{y_i})$. Cette dernière inégalité contredit la définition de **maximize**, ce qui montre que $\widehat{v}_{(n)}(\overrightarrow{\mathbf{u}})$ est une solution leximin-optimale, prouvant ainsi la proposition 5.4. ▲

Cette transcription de la notion de tri des variables objectif par le biais de la contrainte de cardinalité n'est pas complètement nouvelle. Plus précisément, et de manière intéressante, cette approche est à la base du développement des algorithmes de filtrage de [Bleuzen-Guernalec et Colmerauer, 1997] et de [Mehlhorn et Thiel, 2000] pour la contrainte **Sort**, comme on nous l'a fait remarquer [1]. Les deux algorithmes présentés dans ces papiers s'appuient sur les travaux [Zhou, 1997], qui introduit un mécanisme de propagation de la contrainte de tri fondé sur la même idée que les contraintes **AtLeast**. Cette approche a cependant un intérêt particulier dans le cadre de l'optimisation leximin, car elle permet d'introduire les variables du vecteur trié au fur et à mesure de l'algorithme.

Propagation de la contrainte AtLeast À cause de sa généricité, la méta-contrainte **AtLeast** ne peut pas fournir de procédures de filtrage très efficaces. Heureusement, dans notre cas pour lequel toutes les contraintes de Γ sont de la forme $\mathbf{y} \leq \mathbf{x_i}$, la cohérence de borne peut être assurée à l'aide d'une procédure simple, présentée dans l'algorithme 7 (nous rappelons que la notation $\overline{\mathbf{x}} \leftarrow \alpha$ signifie que toutes les valeurs supérieures à α sont enlevées de $\mathscr{D}_{\mathbf{x}}$).

Algorithme 7 — Calcul de la cohérence de borne sur la méta-contrainte **AtLeast** portant sur des contraintes linéaires.

entrée : Un réseau de contraintes $(\mathscr{X}, \mathcal{D}, \mathscr{C})$, vecteur de variables $(\mathbf{x_1}, \dots, \mathbf{x_n}) \in \mathscr{X}^n$, une variable $\mathbf{y} \in \mathscr{X}$, un entier $k \leq n$.
sortie : $bc(\mathbf{AtLeast}(\{\mathbf{y} \leq \mathbf{x_1}, \dots, \mathbf{y} \leq \mathbf{x_n}\}, k))$, ou «Inconsistant».

1 **si** $\sum_i (\underline{\mathbf{y}} \leq \overline{\mathbf{x_i}}) < k$ **retourner** *«Inconsistant»*;
2 **if** $\sum_i (\underline{\mathbf{y}} \leq \overline{\mathbf{x_i}}) = k$ **then**
3   **pour tous les** j **tels que** $\underline{\mathbf{y}} \leq \overline{\mathbf{x_j}}$ **faire** $\underline{\mathbf{x_j}} \leftarrow \underline{\mathbf{y}}$;
4 $\overline{\mathbf{y}} \leftarrow \overrightarrow{\overline{\mathbf{x}}}^{\downarrow}_k$;   /* où $\overrightarrow{\overline{\mathbf{x}}} = (\overline{\mathbf{x_1}}, \dots, \overline{\mathbf{x_n}})$ */
5 **retourner** $(\mathscr{X}, \mathcal{D}, \mathscr{C})$;

De manière informelle, cet algorithme fonctionne comme suit. Si les domaines des variables sont tels que la contrainte ne peut plus être satisfaite (ligne 1), la procédure renvoie «Inconsistant». Sinon, si seulement k variables parmi les variables $\mathbf{x_i}$ peuvent encore être supérieures à $\mathbf{y}$, alors ces

[1] Cette remarque nous a été signalée par Diego Olivier Fernandez Pons au cours d'une discussion.

variables ne peuvent être inférieures à $\underline{\mathbf{y}}$, sinon la contrainte est violée (ligne 3). Dans tous les cas, la valeur de $\underline{\mathbf{y}}$ ne peut être plus grande que la $k^{\text{ème}}$ plus grande valeur des variables $\overline{\mathbf{x_i}}$.

Cet algorithme s'exécute en temps $O(n)$, puisque la sélection de la $k^{\text{ème}}$ plus grande valeur de $\overrightarrow{\overline{\mathbf{x}}}$ peut s'effectuer en $O(n)$ [Cormen *et al.*, 2001, page 189]. On peut de plus remarquer que cet algorithme est bien adapté à une implantation événementielle de la propagation de contraintes : dans le cas d'une mise à jour de l'un des $\overline{\mathbf{x_i}}$, seules les lignes 2–4 ont besoin d'être exécutées (car la mise à jour de $\overline{\mathbf{y}}$ se chargera de vider le domaine de $\mathbf{y}$ si la condition de la ligne 1 n'est plus satisfaite) ; dans le cas d'une mise à jour de $\underline{\mathbf{y}}$, seules les lignes 1–3 ont besoin d'être exécutées ; enfin, tout autre mise à jour ne nécessite aucune exécution de l'algorithme. La procédure peut aussi bénéficier du stockage du vecteur ordonné $\overrightarrow{\overline{\mathbf{x}}}^{\downarrow}$ et de sa mise à jour lorsque l'un des $\overline{\mathbf{x_i}}$ change, mise à jour qui nécessite un temps $O(n)$. Cette opération nous permet d'accéder à $\overrightarrow{\overline{\mathbf{x}}}^{\downarrow}_k$ en temps $O(1)$.

Nous pouvons noter que puisque toutes les contraintes en argument de la méta-contrainte **AtLeast** sont linéaires, cette méta-contrainte peut aussi être exprimée par un ensemble de contraintes linéaires, ce qui rend notre algorithme implantable dans un solveur linéaire (si toutes les autres contraintes sont linéaires). La méthode classique [Garfinkel et Nemhauser, 1972, page 11] consiste à exprimer notre contrainte **AtLeast** en introduisant n variables 0–1 $\{\delta_1, \ldots, \delta_n\}$, ainsi qu'un ensemble de contraintes linéaires $\{\mathbf{y} \leq \mathbf{x_1} + \delta_1 \overline{\mathbf{y}}, \ldots, \mathbf{y} \leq \mathbf{x_n} + \delta_n \overline{\mathbf{y}}, \sum_{i=1}^{n} \delta_i \leq n - k\}$.

5.4.3.4 Transformations max-min

Il existe un autre moyen de faire apparaître la version triée du vecteur objectif, sans toutefois utiliser des contraintes spécifiques associées à leur mécanisme de propagation comme le font les deux algorithmes précédents : on peut utiliser un ensemble de «transformations max-min». Cette solution, introduite dans [Maschler *et al.*, 1992] (et citée dans [Ogryczak, 1997]) afin de traiter des problèmes d'optimisation leximin sur des ensembles non convexes d'alternatives, est fondée sur l'idée suivante : si l'on remplace deux composantes $\mathbf{u_i}$ et $\mathbf{u_j}$ du vecteur objectif par deux variables $\mathbf{m}$ et $\mathbf{M}$ qui représentent respectivement le minimum et le maximum de ces deux composantes, on ne change pas l'ensemble des solutions leximin-optimales.

Proposition 5.5 *Soient* $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ *un réseau de contraintes,* $\overrightarrow{\mathbf{u}}$ *un vecteur objectif, et* $\mathbf{u_i}$ *et* $\mathbf{u_j}$ *deux variables distinctes de* $\overrightarrow{\mathbf{u}}$. *Soient aussi* $\mathbf{m}$ *et* $\mathbf{M}$ *deux nouvelles variables, et* $(\mathscr{X}', \mathcal{D}', \mathscr{C}')$ *le réseau de contraintes défini de la manière suivante :* $\mathscr{X}' = \mathscr{X} \cup \{\mathbf{m}, \mathbf{M}\}$, $\mathcal{D}' = \langle \mathcal{D}, (\mathbf{m} : \mathscr{D}_{\mathbf{m}}, \mathbf{M} : \mathscr{D}_{\mathbf{M}}) \rangle$, *avec* $\mathscr{D}_{\mathbf{m}} = \mathscr{D}_{\mathbf{M}} = [\![\min(\underline{\mathbf{u_i}}, \underline{\mathbf{u_j}}), \max(\overline{\mathbf{u_i}}, \overline{\mathbf{u_j}})]\!]$, *et* $\mathscr{C}' = \mathscr{C} \cup \{\mathbf{m} = \mathbf{Min}(\mathbf{u_i}, \mathbf{u_j}), \mathbf{M} = \mathbf{Max}(\mathbf{u_i}, \mathbf{u_j})\}$. *Nous définissons de même* $\overrightarrow{\mathbf{u}'}$ *comme étant le vecteur égal à* $\overrightarrow{\mathbf{u}}$ *sur toutes ses composantes, excepté* $\mathbf{u_i}$ *qui* $\mathbf{u_j}$ *qui sont remplacées par* $\mathbf{m}$ *et* $\mathbf{M}$.

Nous avons de manière évidente $maxleximin(\mathscr{X}, \mathcal{D}, \mathscr{C}, \overrightarrow{\mathbf{u}}) = maxleximin(\mathscr{X}', \mathcal{D}', \mathscr{C}', \overrightarrow{\mathbf{u}'})_{\downarrow \mathscr{X}}$.

En appliquant de manière récursive cette règle de reformulation, nous pouvons remplacer le vecteur objectif $\overrightarrow{\mathbf{u}}$ en introduisant des nouvelles variables $\overrightarrow{\mathbf{u}'}$ et les contraintes suivantes :

▷ $\forall i \in [\![1, n-1]\!]$, $\mathbf{u'_i} = \mathbf{Max}(\mathbf{Min}\{\mathbf{u_1}, \ldots, \mathbf{u_i}\}, \mathbf{u_{i+1}})$,

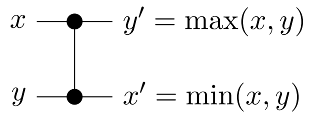
▷ $\mathbf{u'_n} = \mathbf{Min}\{\mathbf{u_1}, \ldots, \mathbf{u_n}\}$.

En utilisant cette reformulation, le minimum des variables objectif apparaît de manière naturelle comme une nouvelle variable, et cette variable constitue clairement le seul sous-ensemble saturé de cardinalité minimale parmi les variables du vecteur objectif. Tout comme dans l'algorithme 2, cette variable est fixée à sa valeur maximale $\widehat{m}$, puis enlevée de l'ensemble des variables objectif.

Il est très intéressant de remarquer que se cachent derrière les transformations max-min une interprétation de l'algorithme de tri fondée sur les réseaux de comparaisons [Cormen *et al.*, 2001, page 704]. Les réseaux de comparaisons sont introduits en tant que modèles de calcul n'utilisant qu'un seul opérateur, l'opérateur de comparaison :

Définition 5.12 (Comparateur) *Un* comparateur *est un dispositif possédant deux entrées numériques x et y et deux sorties x' et y', et qui effectue les fonctions suivantes :* $x' = \max(x, y)$ *et* $y' = \min(x, y)$.

Un comparateur est représenté graphiquement de la manière suivante :



Un certain nombre d'algorithmes de tri de tableaux (ou vecteurs) ont été adaptés au modèle des réseaux de comparaisons. L'algorithme d'optimisation leximin qui a été proposé dans [Maschler *et al.*, 1992] et que nous introduisons ici utilise de manière implicite l'un de ces algorithmes de tri : chaque utilisation d'un comparateur correspond à une reformulation min-max. Cet algorithme de tri est une adaptation de l'algorithme de tri à bulles pour les réseaux de comparaisons, et est représenté de manière graphique dans la figure 5.4.

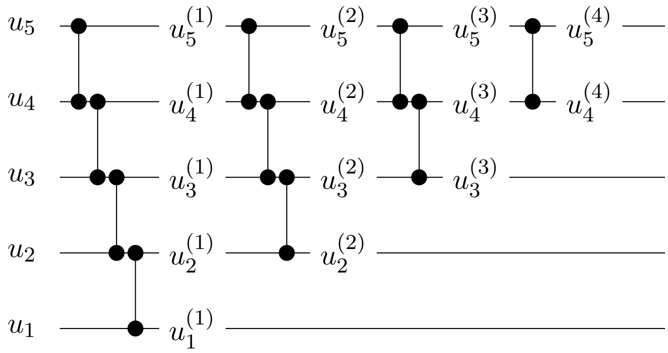


Figure 5.4 — Le réseau de comparaisons correspondant à l'algorithme de tri utilisé de manière implicite dans l'algorithme 8 pour $n = 5$.

Chaque comparateur de la figure 5.4 est «implanté» par deux contraintes dans l'algorithme 8, et chaque point correspond à une variable différente. Noter que les variables et les contraintes sont introduites couche par couche, puisqu'à chaque pas nous n'avons besoin que de la variable minimale (la variable $\mathbf{u_i^{(i)}}$ dans la figure et dans l'algorithme). Les couches sont introduites à chaque pas par la fonction `MinLayer`. Rappelons de plus qu'avant d'introduire une nouvelle couche, on doit restreindre l'ensemble des solutions admissibles à celles telles que la variable objectif minimum est maximale. C'est le rôle de la ligne 9 de l'algorithme 8.

Exemple 5.1.e Nous allons illustrer le fonctionnement de l'algorithme 8 sur l'exemple 5.1. L'algorithme fonctionne en substituant successivement les variables objectif, afin de faire apparaître la variable maximin de manière naturelle.

Le tableau ci-dessous liste l'ensemble des solutions pour les variables objectif initiales et pour les nouvelles. Au premier pas de l'algorithme, on introduit les variables $\overrightarrow{\mathbf{u^{(1)}}}$, et on maximise $\mathbf{u_3^{(1)}}$. On fixe cette dernière variable à sa valeur maximale, ce qui a pour effet de restreindre l'ensemble des solutions (ce qui explique les cellules vides dans le tableau). Au deuxième pas de l'algorithme, on introduit les variables $\overrightarrow{\mathbf{u^{(2)}}}$, et on maximise $\mathbf{u_2^{(2)}}$. Enfin, on introduit et on maximise $\mathbf{u_1^{(3)}}$ au tout dernier pas de l'algorithme.

Algorithme 8 — Résolution du problème [MAXLEXIMINCSP] par transformations max-min.

entrée : Un réseau de contraintes $(\mathscr{X}, \mathcal{D}, \mathscr{C})$; un vecteur objectif $(\mathbf{u_1}, \ldots, \mathbf{u_n}) \in \mathscr{X}^n$.
sortie : Une solution au problème [MAXLEXIMINCSP].

si $\mathbf{solve}(\mathscr{X}, \mathcal{D}, \mathscr{C}) =$ *«Inconsistant»* **alors retourner** *«Inconsistant»*;
$(\mathscr{X}', \mathcal{D}', \mathscr{C}') \leftarrow (\mathscr{X}, \mathcal{D}, \mathscr{C})$;
$\overrightarrow{\mathbf{u}}^{(\mathbf{0})} \leftarrow \overrightarrow{\mathbf{u}}$;
pour $i \leftarrow 1$ **à** n **faire**
  $\mathscr{X}' \leftarrow \mathscr{X}' \cup \{\mathbf{y_i^{(i)}}, \ldots, \mathbf{y_n^{(i)}}\} \cup \{\mathbf{u_i^{(i)}}, \ldots, \mathbf{u_n^{(i)}}\}$;
  $\mathcal{D}' \leftarrow \langle \mathcal{D}', (\mathbf{y_i^{(i)}} : \mathscr{D}_{\mathbf{y_i^{(i)}}}, \ldots, \mathbf{y_n^{(i)}} : \mathscr{D}_{\mathbf{y_n^{(i)}}}), (\mathbf{u_i^{(i)}} : \mathscr{D}_{\mathbf{u_i^{(i)}}}, \ldots, \mathbf{u_n^{(i)}} : \mathscr{D}_{\mathbf{u_n^{(i)}}}) \rangle$ avec $\mathscr{D}_{\mathbf{u_j^{(i)}}} = \mathscr{D}_{\mathbf{y_j^{(i)}}} = [\![\min_k(\underline{\mathbf{u_k}}), \max_k(\overline{\mathbf{u_k}})]\!]$;
  $\mathscr{C} \leftarrow \mathscr{C} \cup \mathtt{MinLayer}(\overrightarrow{\mathbf{u}}^{(\mathbf{i-1})}, \overrightarrow{\mathbf{u}}^{(\mathbf{i})}, \overrightarrow{\mathbf{y}}^{(\mathbf{i})})$;
  $\widehat{v}_{(i)} \leftarrow \mathbf{maximize}(\mathscr{X}', \mathcal{D}', \mathscr{C}', \mathbf{u_i^{(i)}})$;
  $\mathbf{u_i^{(i)}} \leftarrow \widehat{v}_{(i)}(\mathbf{u_i^{(i)}})$;
retourner $\widehat{v}_{(n)\downarrow\mathscr{X}}$;

Fonction `MinLayer`$(\{\mathbf{u_1}, \ldots, \mathbf{u_m}\}, \{\mathbf{v_1}, \ldots, \mathbf{v_m}\}, \{\mathbf{y_1}, \ldots, \mathbf{y_m}\})$

entrée : Trois vecteurs de taille m.
sortie : Un ensemble de contraintes.

$\mathscr{C} \leftarrow \{\mathbf{y_m} = \mathbf{u_m}\}$;
pour $i \leftarrow m$ **à** 2 **faire**
  $\mathscr{C} \leftarrow \mathscr{C} \cup \{\mathbf{v_i} = \mathbf{Max}(\mathbf{y_i}, \mathbf{u_{i-1}}), \mathbf{y_{i-1}} = \mathbf{Min}(\mathbf{y_i}, \mathbf{u_{i-1}})\}$;
$\mathscr{C} \leftarrow \{\mathbf{v_1} = \mathbf{y_1}\}$;
retourner $\mathscr{C}$;

$\mathbf{u_1}$	$\mathbf{u_2}$	$\mathbf{u_3}$	$\mathbf{u_1^{(1)}}$	$\mathbf{u_2^{(1)}}$	$\mathbf{u_3^{(1)}}$	$\mathbf{u_1^{(2)}}$	$\mathbf{u_2^{(2)}}$	$\mathbf{u_1^{(3)}}$
1	1	0	1	1	0			
5	5	3	5	5	**3**	5	**5**	5
7	3	5	7	5	**3**	7	**5**	**7**
1	2	1	2	1	1			
9	5	2	9	5	2			
3	4	3	4	3	**3**	4	3	
5	3	6	5	6	**3**	6	**5**	6
10	3	4	10	4	**3**	10	4	

5.4.4 Aspects heuristiques

Les algorithmes de programmation par contraintes dédiés au problème [MAXLEXIMINCSP] peuvent bénéficier d'une heuristique d'instanciation de variables spécifique qui tire partie de la sémantique particulière du leximin pour guider la recherche arborescente vers de meilleures solutions. En effet, lors de la recherche d'une solution leximin-optimale, la composante la plus basse du vecteur objectif est déterminante, car c'est elle qui doit être augmentée en premier pour obtenir une meilleure solution. Ceci nous donne une idée très simple d'heuristique : la prochaine variable à instancier devra augmenter le plus possible la plus basse valeur du vecteur objectif.

Dans les problèmes de décision collective ou d'allocation de ressources, le vecteur objectif, qui est le profil d'utilités, dépend en général de variables de décision (typiquement les variables 0–1 correspondants aux variables booléennes $\mathbf{alloc(o, i)}$). Dans ce contexte, la prochaine variable à instancier devra être la variable de décision qui fait le plus augmenter l'utilité de l'agent le moins satisfait au moment de la décision.

Le but d'une telle heuristique est de faire tendre le vecteur objectif relativement vite vers une solution assez égalitaire, pour laquelle il ne sera alors plus possible d'augmenter l'utilité du moins satisfait des agents. Dans ce cas, on devra ensuite chercher à augmenter l'utilité du deuxième agent dans l'ordre croissant d'utilités. Ainsi de suite jusqu'au dernier agent.

Naturellement, ces considérations ne donnent qu'une piste pour la définition d'une heuristique, qui doit être déclinée pour chaque instance traitée. On pourra prendre en compte d'autres considérations que l'utilité de l'agent le moins satisfait, comme par exemple la taille d'un lot à attribuer dans un problème d'allocations de ressources, ou le rapport du gain d'utilité espéré pour les agents sur la consommation de la ressource. Comme nous le verrons brièvement au chapitre 6 consacré aux expérimentations, ces aspects heuristiques sont cruciaux pour l'efficacité des algorithmes.

5.5 Conclusion : au-delà du leximin ?

Nous avons introduit dans ce chapitre un certain nombre d'approches du problème de calcul d'une solution leximin-optimale dans un réseau de contraintes. Nous avons justifié l'intérêt de se pencher sur ce problème, tout d'abord en rappelant l'ensemble des bonnes propriétés du préordre leximin pour l'agrégation des utilités des agents, et dans un deuxième temps en mettant en valeur la relative difficulté algorithmique de ce problème, et l'inadéquation de l'approche à base de transcription du préordre sous la forme d'une fonction d'utilité collective.

Nous nous devons cependant de tempérer ces commentaires sur le préordre leximin, car il peut poser quelques problèmes dans certaines situations concrètes que nous allons brièvement exposer. Rappelons tout d'abord que le préordre leximin a été introduit comme un raffinement de l'ordre social induit par la fonction min pour pallier les problèmes liés à l'idempotence de l'opérateur min : dans un problème à n agents, les profils d'utilité $(0, \ldots, 0)$ et $(0, 1000, \ldots, 1000)$ sont indifférents pour l'ordre social induit par la fonction min (car ils produisent la même utilité 0).

L'ordre leximin permet d'éviter ce type de problèmes en permettant de distinguer selon les composantes suivantes plusieurs profils qui ont un minimum identique. Cependant, dans de nombreux problèmes de partage, l'ordre leximin peut sembler presque aussi extrême que l'ordre induit par min, car il n'autorise aucune concession sur la composante minimum des profils d'utilité, et ce même si une légère diminution de cette composante minimum permettait d'augmenter de manière considérable les utilités des autres agents. Considérons par exemple les deux profils d'utilité suivants : $(10, \ldots, 10)$ et $(9, 1000, \ldots, 1000)$. Entre ces deux profils, le préordre leximin (tout comme le min) préférera le premier, alors qu'il peut sembler plus naturel dans beaucoup de cas de faire une concession sur la valeur minimale si les autres utilités sont vraiment meilleures[2]. En d'autres termes, il n'y a aucune compensation possible entre une infime perte d'utilité de l'agent le moins heureux et un immense gain sur les autres agents.

Cette absence de compensation soulève un autre problème pratique, liée au calcul d'une solution leximin-optimale. Lorsque l'on se place dans le contexte de la programmation par contrainte, où les domaines des variables sont finis, l'utilisation des algorithmes décrits dans ce chapitre ne pose pas de problème de correction. En revanche, il en est autrement lorsque l'on se place dans le cadre du calcul flottant, comme c'est le cas lorsque l'on traite le problème à l'aide d'un solveur linéaire (comme nous le ferons en partie dans le chapitre 6). À chaque étape de l'algorithme, une erreur

[2]Remarque : aurait-on accepté de faire cette concession si les deux profils avaient été $(1, 10, \ldots, 10)$ et $(0, 1000, \ldots, 1000)$?

d'approximation sur une composante du leximin peut avoir des répercussions énormes sur le reste du calcul : soit les composantes du vecteur sont très différentes de ce qu'elles devraient être en vertu de l'absence de compensation possible, soit l'algorithme s'arrête sur une incohérence aux étapes suivantes.

Prenons un exemple : considérons un problème pour lequel il existe deux profils d'utilité admissibles $(9, 10, 10, 10, 10)$ et $(9, 9.9, 1000, 1000, 1000)$. Supposons qu'au moment du calcul de la deuxième composante du leximin, le solveur fasse une erreur numérique et trouve 9.9 au lieu de 10. Cela a une immense répercussion sur la suite du vecteur, puisque les composantes suivantes seront égales à 1000, alors qu'elles devraient être égales à 100 si l'algorithme était correct. Cet exemple met en évidence la grande sensibilité du préordre leximin vis-à-vis des composantes faibles du vecteur objectif.

La mise en évidence de ces deux problèmes pose donc la question de la pertinence du préordre leximin dans les problèmes d'agrégation d'utilités. En d'autres termes, peut-on trouver un ordre social moins «drastique» que l'ordre leximin, c'est-à-dire qui admette des compromis sur la perte d'utilité des agents les moins riches ?

5.5.1 Un leximin à seuil

La notion d'égalité est relativement mal définie dans le domaine du calcul flottant. Dans ce contexte, il est nécessaire d'introduire une certaine tolérance aux approximations, en considérant que deux nombres sont égaux si leur différence est inférieure à un certain seuil représentant la précision machine (certains langages ou systèmes de calcul introduisent ce seuil de manière explicite : voir par exemple la variable `eps` dans le logiciel de calcul numérique *Matlab*).

De manière plus générale, un agent sera souvent indifférent entre deux utilités très proches, mais ne le sera pas si les utilités sont plus éloignées. Comme nous l'avons vu au chapitre 1, cette remarque est à la base de la définition des préférences de type semi-ordres, qui font intervenir un seuil d'indifférence q en deçà duquel la différence entre deux utilités n'est plus pertinente.

Peut-on adapter l'idée des préférences à seuil au préordre leximin ? D'un point de vue purement formel, la réponse est positive, et la notion de leximin à seuil peut être définie comme suit. Soient deux profils d'utilité $\overrightarrow{u}$ et $\overrightarrow{v}$, et $\varepsilon > 0$. Alors les deux relations $\prec_{leximin}^{(\varepsilon)}$ et $\sim_{leximin}^{(\varepsilon)}$ dont les suivantes :

$$\begin{cases} \overrightarrow{u} \prec_{leximin}^{(\varepsilon)} \overrightarrow{v} & \Leftrightarrow \quad \exists i \in [\![1, n]\!] \text{ t.q. } \begin{cases} \forall j < i, |u_j^{\uparrow} - v_j^{\uparrow}| \leq \varepsilon, \text{ et} \\ u_i^{\uparrow} > v_i^{\uparrow} + \varepsilon, \end{cases} \\ \overrightarrow{u} \sim_{leximin}^{(\varepsilon)} \overrightarrow{v} & \Leftrightarrow \quad \forall i \in [\![1, n]\!], \ |u_i^{\uparrow} - v_i^{\uparrow}| \leq \varepsilon. \end{cases}$$

La simplicité de cette extension cache néanmoins quelques problèmes. Le premier de ces problèmes est que cette relation n'est plus compatible avec la relation de dominance de Pareto qu'au sens faible (tout comme la fonction min). Considérons par exemple le cas où $\varepsilon = 10$, et les deux profils d'utilité suivants : $\overrightarrow{u} = (10, 10, 10)$ et $\overrightarrow{v} = (15, 15, 15)$. On a $\overrightarrow{u} \sim_{leximin}^{(\varepsilon)} \overrightarrow{v}$ alors que $\overrightarrow{u}$ domine $\overrightarrow{v}$ au sens de Pareto. Ce problème n'est pas vraiment rédhibitoire : à l'issu du processus de choix collectif, il suffit d'éliminer les décisions non Pareto-efficaces.

La relation leximin à seuil pose un autre problème légèrement plus sérieux. Considérons par exemple le cas où $\varepsilon = 10$, et introduisons les trois profils d'utilité suivants : $\overrightarrow{u} = (10, 30, 30)$, $\overrightarrow{v} = (10, 15, 70)$ et $\overrightarrow{w} = (10, 22, 50)$. Nous avons : $\overrightarrow{u} \prec_{leximin}^{(\varepsilon)} \overrightarrow{v}$ (la deuxième valeur est discriminante), $\overrightarrow{v} \prec_{leximin}^{(\varepsilon)} \overrightarrow{w}$ (la troisième valeur est discriminante), et $\overrightarrow{w} \prec_{leximin}^{(\varepsilon)} \overrightarrow{u}$ (la troisième valeur est discriminante). Nous voyons donc que la relation $\prec_{leximin}^{(\varepsilon)}$ est non transitive, ce qui revient à dire — car il s'agit d'un préordre total — qu'elle est cyclique. La question est de choisir quelle alternative choisir dans ce cas-là. Cette question n'est pas nouvelle dans le domaine de la représentation et de l'agrégation de préférences, et il existe des solutions pour traiter ce problème de cyclicité de la relation de préférence [Vincke, 1989]. Nous pouvons cependant nous interroger sur la pertinence (et la complexité) de cette approche, sachant que le cadre du *welfarisme* cardinal nous fournit d'autres moyens d'introduire des concessions sur le préordre leximin, comme nous allons le voir.

5.5.2 Les moyennes pondérées ordonnées

Les moyennes pondérées ordonnées (OWA), qui ont été introduites dans le chapitre 1 en tant que famille de fonctions d'utilité collective, constituent un relâchement assez naturel du leximin (qui présente aussi l'avantage d'être facilement paramétrable selon l'importance de la concession que l'on veut faire par rapport au leximin). L'ensemble des OWA équitables est l'ensemble des OWA dont le vecteur de coefficients $\overrightarrow{w}$ est tel que $i > j \Rightarrow w_i < w_j$: c'est l'ensemble des OWA qui respectent le principe de réduction des inégalités.

Nous pouvons nous interroger sur la transcription des algorithmes de résolution du problème [MAXLEXIMINCSP] au problème de maximisation d'une fonction d'utilité collective de type OWA. La plupart des algorithmes introduits dans ce chapitre sont fondés sur la propriété d'absence de compensation entre les utilités des agents les moins riches et des agents les plus riches, qui implique que chaque composante du vecteur leximin-optimal peut être calculée séparément, ce qui n'est plus le cas pour les OWA.

En revanche, une grande partie du travail effectué autour de ces algorithmes concerne la définition de la notion de tri d'un vecteur sous forme de contrainte, et les algorithmes de propagation dédiés à cette notion de tri. Cette contrainte de tri est bien évidemment indispensable pour la définition de la variable d'utilité collective, définie par une moyenne pondérée ordonnée. Nous pouvons donc bénéficier de la contrainte **Sort**, ou encore de la définition du vecteur objectif trié par l'introduction de contraintes **Max** et **Min**, couplées avec une contrainte linéaire sur le vecteur objectif trié, afin de définir la fonction d'utilité collective OWA.

Notons que l'on trouve dans la littérature un article qui traite de moyennes pondérées ordonnées «équitables» dans un contexte de recherche opérationnelle : [Ogryczak et Śliwiński, 2003]. Cet article propose une modélisation sous forme de programme linéaire du problème de maximisation d'un OWA sous contraintes linéaires. Il pourrait être intéressant de comparer cette approche avec l'approche programmation par contraintes fondée sur la contrainte **Sort**.

Chapitre 6

Expérimentations

Nous avons introduit au chapitre précédent plusieurs algorithmes dédiés à l'optimisation du préordre leximin dans le cadre de la programmation par contraintes. Nous allons, dans ce chapitre, présenter les résultats expérimentaux obtenus sur la comparaison de ces algorithmes sur des instances de problèmes de partage combinatoires générées artificiellement ou réelles.

L'intérêt de ce travail expérimental est double. D'une part il permet de mettre en valeur les caractéristiques opérationnelles des algorithmes introduits dans le chapitre 5, en comparant leurs efficacités respectives sur des instances très différentes. D'autre part, il nous a permis de travailler sur la génération d'instances réalistes de problèmes de partage. Cette problématique est cruciale dans le domaine du partage et plus généralement du choix social, car, la discipline n'ayant été que récemment étudiée par les chercheurs en informatique et intelligence artificielle, peu de travaux abordent le problème de la génération d'instances réalistes aléatoires dans ce contexte. On trouve toutefois une exception dans le domaine des enchères combinatoires [Leyton-Brown *et al.*, 2000], dont nous allons parler à la section 6.2.

Dans ce chapitre, nous allons introduire les quatre problèmes de partage particuliers sur lesquels nous avons travaillé, sous l'angle de la modélisation, de la génération d'instances, et de la comparaison des algorithmes dédiés au calcul de solutions leximin-optimales. Ces problèmes sont les suivants :

- ▷ **Le problème Pléiades simplifié :** il s'agit d'une version simplifiée du problème de partage de la constellation de satellites Pléiades, dans laquelle les contraintes opérationnelles sont approchées par des contraintes linéaires, et les préférences des agents représentées dans un langage additif.
- ▷ **Les enchères combinatoires équitables :** il s'agit d'une adaptation du *Winner Determination Problem* dans les enchères combinatoires, adaptation dans laquelle le critère est le préordre leximin et non la somme comme dans le problème classique.
- ▷ **Le problème de partage de biens indivisibles général :** il s'agit du problème de partage de biens indivisibles tel qu'il a été introduit au chapitre 3 dans la définition 3.38 (préférences et contraintes spécifiées dans un langage à base de logique).
- ▷ **Problème d'affectation de sujets de travaux expérimentaux :** il s'agit d'une instance réelle d'un problème d'affectation de sujets à des élèves.

Pour tous ces problèmes, nous avons travaillé sur des instances créées artificiellement, sauf pour le dernier d'entre eux, pour lequel nous disposons d'une instance réelle. Nous présenterons, pour chacune de ces applications, une description formelle du problème dans un premier temps, puis suivra un aperçu de la manière dont sont générées les instances, et enfin des résultats obtenus sur la comparaison expérimentale des algorithmes du chapitre 5 sur ces instances.

Spécifications techniques : Toutes les implantations logicielles décrites dans ce chapitre ont été effectuées dans le langage Java version 1.5.0. La bibliothèque de programmation par contraintes utilisée pour la résolution des problèmes est la bibliothèque Choco [Laburthe, 2000], et nous avons de même utilisé la bibliothèque ILOG CPLEX [ILOG, 2006] pour la résolution de certains problèmes linéaires. Les expérimentations ont été exécutées sur une station Sun équipée d'un processeur `SUNW,UltraSPARC-IIIi` tournant à 1,6GHz, d'une mémoire vive de 1Go et du système d'exploitation Solaris 10. Bien entendu, les temps de calcul donnés dans ce chapitre ne sont pas vraiment significatifs de l'efficacité absolue des algorithmes : ils dépendent de la configuration matérielle et logicielle de la plate-forme utilisée, ainsi que du système de programmation par contraintes lui-même. L'intérêt de ces résultats porte donc sur l'efficacité *relative* des algorithmes, étant donné qu'ils ont été exécutés sur les mêmes instances et dans les mêmes conditions.

6.1 Le problème Pléiades simplifié

6.1.1 Description et modélisation du problème

Afin de tester nos algorithmes de résolution dédiés au problème [MAXLEXIMINCSP] sur un problème simple à modéliser et relativement réaliste, nous avons extrait un problème simplifié d'allocation équitable d'objets à des agents de l'application 1 concernant le partage de la constellation de satellites Pléiades. Cette version simplifiée peut être vue comme un sous-problème du problème 3.38 (avec contraintes de volume), défini de la manière suivante.

▷ Les demandes sont atomiques, c'est-à-dire correspondant à des préférences additives, le poids d'un objet o dans les préférences de l'agent i étant noté $w_{i,o}$.

▷ Les contraintes sont de deux types :
  - Les contraintes de volume particulières à chaque agent : r_o est la *ressource consommée* par l'objet o, et $rmax_i$ est la *consommation de ressource maximale* autorisée pour l'agent i. Ces contraintes de volume permettent de simuler des droits exogènes inégaux sur les agents, en jouant sur l'introduction d'un quota sur la ressource consommée par les demandes plutôt que sur la fonction d'utilité collective elle-même.
  - Les contraintes de volume généralisées $\mathscr{C}_{vol}$: $v_{C,o}$ est le *volume* de l'objet o dans la contrainte de volume généralisé C, et $vmax_C$ est le *volume maximum* dans la contrainte de volume généralisé C. L'objectif de cet ensemble de contraintes est de simuler les contraintes physiques du problème réel qui restreignent l'ensemble des allocations admissibles : sur un intervalle de temps donné, le volume d'images acquises est limité par les capacités d'acquisition et d'agilité du satellite.

Notons que dans ce problème, la contrainte de préemption n'est pas présente, et l'on ne requiert pas la complétude de l'allocation. Ce problème peut être modélisé à l'aide de contraintes linéaires uniquement. Cette modélisation est fondée sur les variables suivantes :

▷ Un ensemble de variables 0–1 $\mathbf{x_{i,o}}$: $\mathbf{x_{i,o}} = 1$ si l'objet o est alloué à l'agent i, et 0 sinon, $o \in \mathscr{O}$, $i \in \mathscr{N}$. Les affectations possibles des variables $\mathbf{x_{i,o}}$ représentent l'ensemble des allocations possibles (parmi lesquelles se trouvent les admissibles).

▷ Un ensemble de variables numériques $\mathbf{u_i}$ représentant les utilités individuelles des agents : $\mathbf{u_i} = \sum_{o \in \mathscr{O}} \mathbf{x_{i,o}} \cdot w_{i,o}$ est l'utilité individuelle de l'agent i, $i \in \mathscr{N}$;

▷ Un ensemble de variables 0–1 $\mathbf{s_o}$: $\mathbf{s_o} = \max_{i \in \mathscr{N}} \mathbf{x_{i,o}} = 1$ si l'objet o est alloué à un agent au moins, et 0 sinon.

Les contraintes d'admissibilité du problème, c'est-à-dire les contraintes de volume particulières et généralisées sont les suivantes :

▷ $\sum_{o \in \mathscr{O}} \mathbf{x_{i,o}} \cdot r_o \leq rmax_i$, pour tout $i \in \mathscr{N}$ (contraintes de consommation de ressources) ;
▷ $\sum_{o \in \mathscr{O}} \mathbf{s_o} \cdot v_{c,o} \leq vmax_c$, pour tout $c \in \mathscr{C}$ (contraintes de volume généralisé).

Nous avons ajouté dans la modélisation du problème la contrainte suivante, dont le but est de cerner les allocations réellement significatives et d'augmenter l'efficacité de la modélisation : $w_{i,o} = 0 \Rightarrow \mathbf{x_{i,o}} = 0$ (un objet ne peut être alloué à un agent qui lui attribue un poids nul).

Nous allons donc nous intéresser au problème d'optimisation suivant :

Problème 17: [PLÉIADES SIMPLIFIÉ]

INSTANCE : Un ensemble d'agents $\mathscr{N}$, un ensemble d'objets $\mathscr{O}$, un ensemble de contraintes de consommation définies par $\{r_o \mid o \in \mathscr{O}\}$ et $\{rmax_i \mid i \in \mathscr{N}\}$, et un ensemble de contraintes de volume généralisées $\mathscr{C}_{vol}$.

SOLUTION : Une solution v de $maxleximin((\mathscr{X}, \mathcal{D}, \mathscr{C}), \overrightarrow{\mathbf{u}})$, avec $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ le réseau de contraintes défini par les variables $\mathbf{x_{i,o}}$, $\mathbf{u_i}$ et $\mathbf{s_o}$ et les contraintes de volume généralisées et de consommation.

On peut remarquer que lorsqu'il n'y a qu'un seul agent et une seule contrainte (de consommation ou de volume), ce problème se réduit au problème suivant :

Problème 12 (rappel): [KNAPSACK] [Garey et Johnson, 1979, page 65]

INSTANCE : Un ensemble fini $\mathscr{S}$, une fonction de taille $s : \mathscr{S} \rightarrow \mathbb{N}$, une taille maximale $S_{max} \in \mathbb{N}$, une fonction de valeur (d'utilité) $u : \mathscr{S} \rightarrow \mathbb{N}$, et un entier K.

QUESTION : Existe-t-il un sous-ensemble $\mathscr{S}' \subseteq \mathscr{S}$ tel que $\sum_{a \in \mathscr{S}'} s(a) \leq S_{max}$ et $\sum_{a \in \mathscr{S}'} u(a) \geq K_{max}$?

S'agissant d'un problème NP-complet, notre problème de décision est donc bien sûr aussi NP-complet.

Plus précisément, on peut remarquer qu'il s'agit d'une généralisation du problème de sac à dos multidimensionnel en variables bivalentes [Vasquez et Hao, 2001], mais avec un critère d'optimisation leximin, au lieu du critère somme.

6.1.2 Génération des instances

Nous avons construit un générateur d'instances aléatoires de ce problème d'allocation équitable.

Ce générateur, écrit en Java, est disponible en ligne à l'URL `http://www.cert.fr/dcsd/THESES/sbouveret/benchmark`. L'algorithme de génération des instances est réglable grâce à quatre groupes de paramètres :
▷ les paramètres généraux : nombre d'agents, nombres d'objets, graine du générateur aléatoire ;
▷ les paramètres de poids des objets ;
▷ les paramètres de contraintes de consommation ;
▷ les paramètres de contraintes de volume généralisé.

Les poids des objets sont générés aléatoirement. Deux types de distributions sont possibles : une distribution uniforme entre 0 et w_{max} et une répartition en différentes classes. La répartition en différentes classes approche les conditions de l'application réelle. Une telle répartition est paramétrée par f_c (par exemple $f_c = 10$), le facteur multiplicatif entre classes, et n_c (par exemple $n_c = 4$), le nombre de classes. Les poids appartenant à la classe $i \in [\![1, n_c]\!]$ sont tirés aléatoirement entre $\frac{1}{2}(f_c)^i$ et $\frac{3}{2}(f_c)^i$. De plus, on s'assure qu'il y a plus de demandes de classes faibles (moins importantes), que de demandes de classes fortes.

Comme indiqué précédemment, les contraintes de consommation permettent de simuler des droits d'accès à la ressource inégaux selon les agents. Dans notre générateur, les droits inégaux dépendent de deux paramètres : le droit de l'agent de plus faible droit r_{min}, et le facteur multiplicatif entre les droits f_d. L'agent i a un droit de $r_{min} \times (f_d)^{i-1}$.

Les paramètres concernant les contraintes de volume généralisé sont les suivants :
▷ l'arité des contraintes n_S ;
▷ le volume maximum autorisé pour chaque contrainte (fixe ou aléatoire) ;
▷ le volume de chaque objet (fixe ou aléatoire).
Le générateur permet aussi d'instancier ces paramètres en spécifiant la dureté des contraintes (ici le rapport du nombre d'objets interdits sur l'arité de la contrainte), celles-ci portant sur n_S objets consécutifs.

On pourra trouver une description plus détaillée sur le générateur d'instances sur la page pointée par l'URL citée ci-avant.

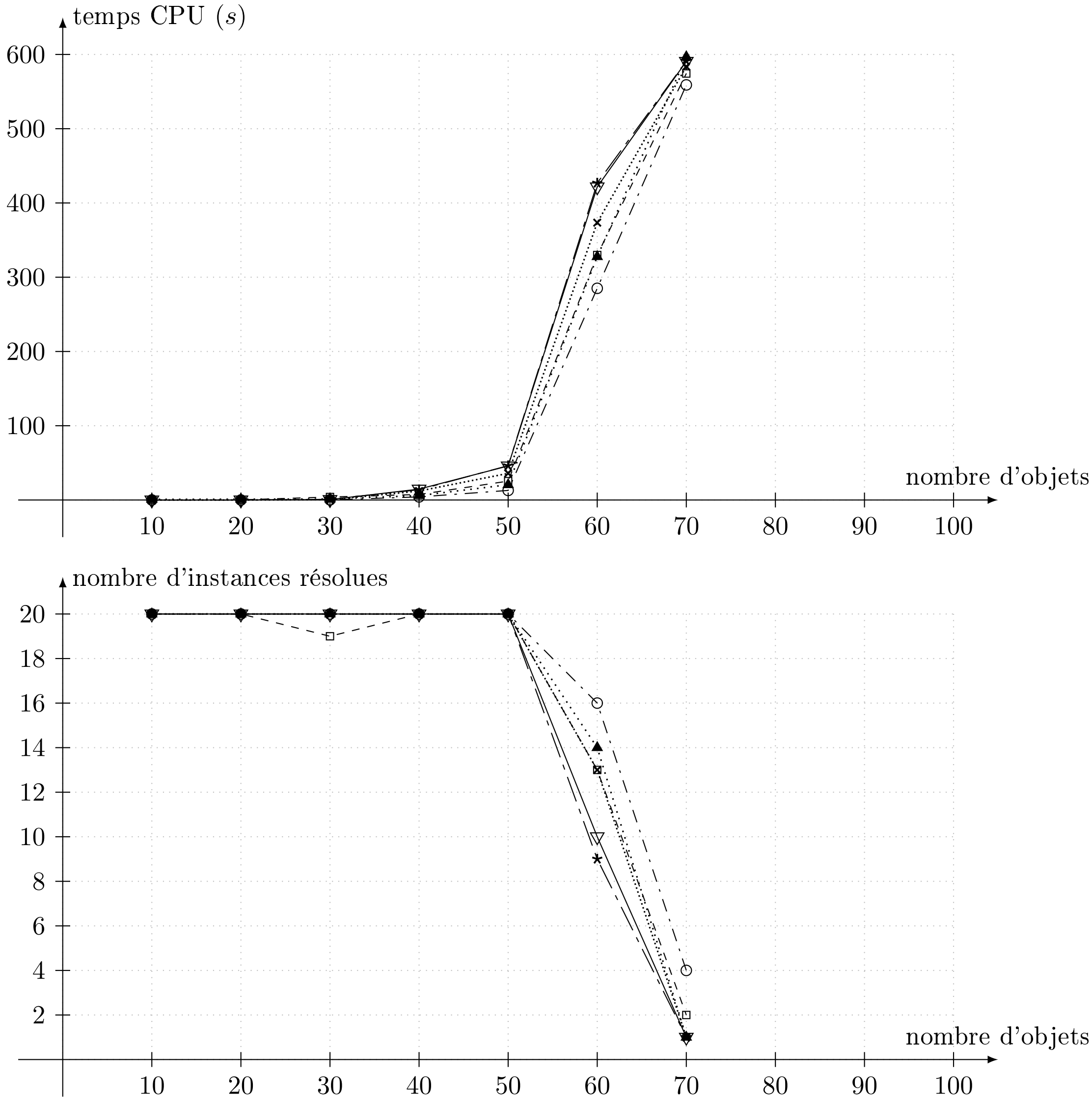
6.1.3 Résultats

Nous avons cherché pour les expérimentations à mettre en valeur l'influence du nombre d'agents et l'influence du nombre d'objets sur l'efficacité des algorithmes. D'un côté le nombre d'agents détermine la taille du vecteur objectif, et de l'autre l'influence du nombre d'objets permet de mettre en valeur le comportement des algorithmes face à la taille des instances, mais surtout de faire varier le type de problème auquel on a affaire.

Nous avons conduit trois séries d'expérimentations sur le problème Pléiades linéaire pour un nombre d'agents respectivement fixé à 4, 10 et 20 et pour un nombre d'objets variable. Dans ces trois séries, les instances ont été générées avec les paramètres par défaut du programme, sauf pour les poids des demandes, dont la répartition est uniforme entre 0 et 100, et pour les droits des agents, qui sont considérés comme égaux (ce qui rend le problème plus difficile en pratique). Les expérimentations ont été conduites sur 20 instances de chaque type donné, le temps de résolution étant limité à 10 minutes par instance. Les courbes présentées dans ce chapitre représentent d'une part le temps moyen d'exécution sur les 20 instances de chaque type, et d'autre part le nombre d'instances résolues dans la limite des 10 minutes. Notons que le temps moyen calculé n'est plus significatif si le nombre d'instances résolues en moins de 10 minutes est trop faible, car le temps de ces instances non résolues est pris en compte dans la moyenne comme étant égal à 10 minutes, ce qui biaise évidemment la moyenne. En conséquence, l'interprétation des courbes de moyenne ne peut se faire séparément de celle des courbes de nombre d'instances résolues.

On peut voir sur la figure 6.1 l'évolution du temps de calcul en fonction du nombre d'objets sur des instances de type Pléiades simplifié à 4 agents. Cette figure ne nous apporte que peu d'information sur l'efficacité relative des algorithmes, car les temps d'exécution de ceux-ci sont quasiment identiques. Cette similarité n'est en fait pas très étonnante. Le problème ne comportant que peu d'agents, il est donc relativement proche d'un problème monoagent sur lequel toutes les approches sont équivalentes (car fondé sur la maximisation de l'utilité individuelle de l'agent en question).

Lorsque l'on augmente le nombre d'agents, on commence à voir apparaître quelques différences entre les algorithmes, comme on peut le remarquer sur la figure 6.2 (10 agents). Sans surprise, l'algorithme fondé sur un calcul et une comparaison explicite de toutes les solutions du problème se révèle largement moins efficace que les autre approches sur ce type d'instances. Notons que l'algorithme 8 fondé sur les transformations max-min ne s'avère pas très efficace non plus ici. Dans le haut du tableau, on pourra remarquer les algorithmes 2 et 1, respectivement fondés sur le calcul des sous-ensembles saturés et sur la contrainte **Leximin**. Nous pouvons remarquer le début d'un



Algorithme 6 fondé sur la contrainte **AtLeast**.
Algorithme 2, inspiré de [Dubois et Fortemps, 1999].
Algorithme fondé sur le calcul et la comparaison de toutes les solutions du problème, inspiré par [Ehrgott, 2000].
Algorithme 8 fondé sur les transformations max-min.
Algorithme 1 fondé sur la contrainte **Leximin**.
Algorithme 5 fondé sur la contrainte **Sort**.

Figure 6.1 — Comparaison des temps de calcul des algorithmes de résolution du problème [MAXLEXIMINCSP] sur des instances du type Pléiades simplifié à 4 agents.

phénomène concernant l'algorithme 2, qui va s'amplifier sur l'exemple suivant : lorsque le nombre d'objets est relativement faible, on note une perte d'efficacité de cet algorithme. L'explication en est relativement simple : lorsque le nombre d'objets est faible et de l'ordre du nombre d'agents, le jeu des contraintes implique l'impossibilité de satisfaire tous les agents, créant ainsi des *ex-aequo* dans les solutions leximin-optimales, rendant d'autant plus difficile le calcul des sous-ensembles saturés de cardinalité minimale (rappelons que la complexité de ce calcul dépend de manière mécanique du nombre d'utilités égales dans le vecteur objectif de la solution leximin-optimale).

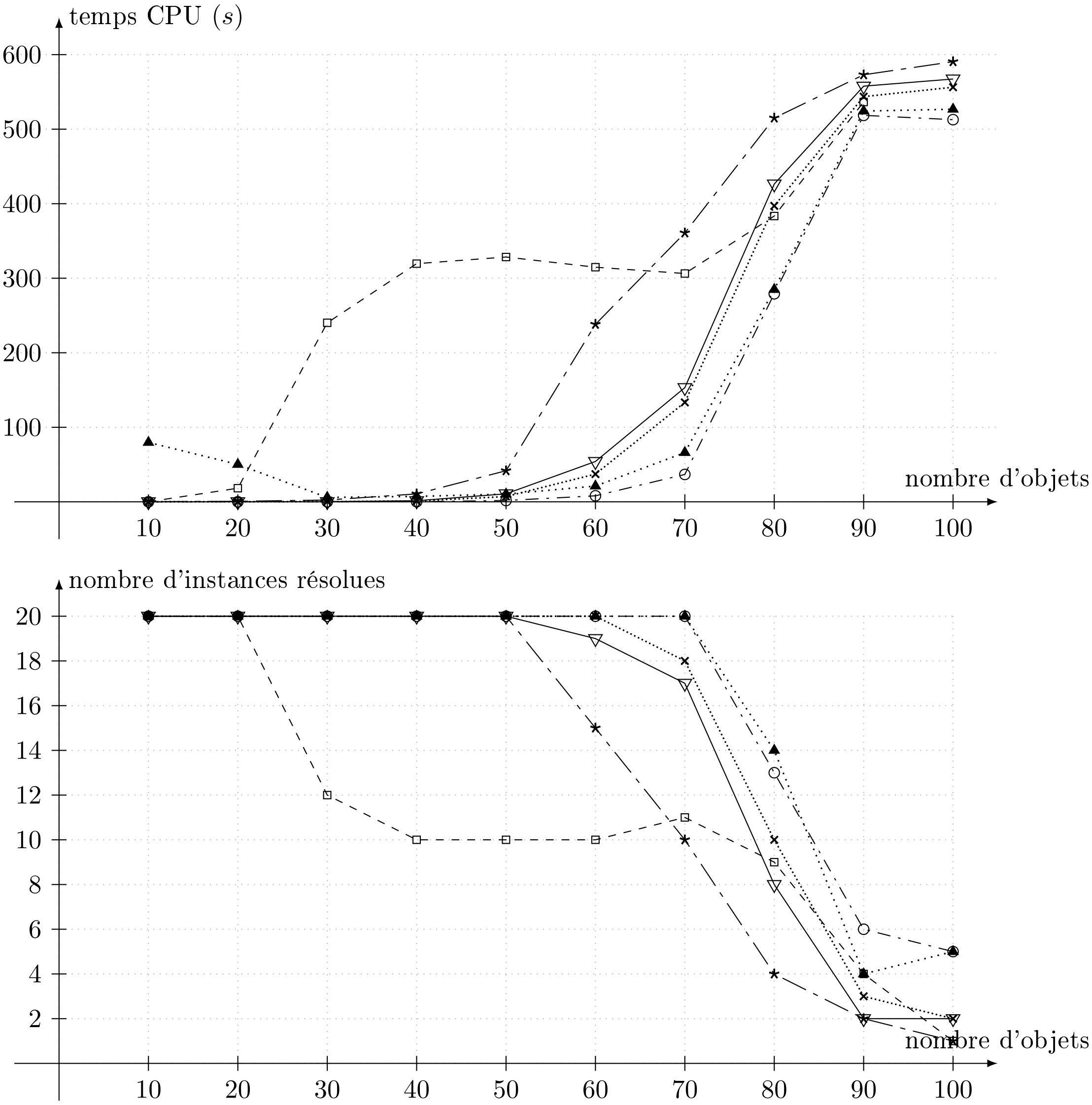
La figure 6.3 (20 agents) est particulièrement intéressante à plusieurs points de vue. D'une part, les écarts entre les algorithmes se creusent, permettant ainsi de les séparer en deux groupes : les algorithmes 1, 5 et 6 semblent relativement efficaces sur ce type d'instances. Le cas de l'algorithme 2 est très particulier : il se révèle complètement inefficace sur les instances à faible nombre d'objets, pour les raisons évoquées ci-avant concernant le calcul des sous-ensembles saturés, mais il semble relativement efficace sur les instances comportant un nombre d'objets plus élevé.

Nous pouvons remarquer sur toutes les courbes précédentes que deux algorithmes donnent des résultats extrêmement similaires, illustrés par la proximité de leurs courbes sur tous les graphes présentés : l'algorithme fondé sur la contrainte **AtLeast** et celui fondé sur la contrainte **Sort**. Cette similarité n'est pas surprenante à la lumière de la remarque que nous avons introduit lors de la description de l'algorithme 6 : ces deux contraintes sont fondées sur la même approche.

Heuristique de choix des variables Nous avons implanté pour les tests une heuristique de choix de variables fondée sur les principes énoncés dans la section 5.4.4 du chapitre précédent (les résultats présentés dans cette section tiennent compte de cette heuristique). Cette heuristique choisit la prochaine variable à instancier de la manière suivante : on choisit, parmi les objets de l'agent actuellement le moins satisfait, l'objet qui a un plus fort poids. L'efficacité de cette heuristique est remarquable. Selon les instances, les temps de calcul peuvent baisser de plus de 50%.

Programmation linéaire Concluons enfin ces commentaires sur les résultats des algorithmes testés sur les instances du modèle Pléiades simplifié en mentionnant le fait que nous avons adapté l'implantation de l'algorithme fondé sur la contrainte **AtLeast** à l'interface permettant de faire la liaison avec le solveur linéaire CPLEX. Nous avons pour cela utilisé la linéarisation de cette dernière contrainte, mentionnée lorsque nous l'avons introduite dans le chapitre 5. Les différences d'efficacité avec l'implantation fondée sur Choco sont flagrantes : dans les instances précédentes, CPLEX peut calculer une solution leximin-optimale en quelques secondes jusqu'à plus de 200 objets. Nous avons cependant rencontré quelques problèmes liés au passage en flottant par CPLEX, provoquant dans certains cas des erreurs de calcul qui, comme nous l'avons mentionné en fin du chapitre 5, peuvent provoquer un arrêt de l'algorithme (dû à une incohérence à une certaine itération), ou peuvent avoir de grandes répercussions sur les valeurs du vecteur leximin-optimal trouvé. La source de ces problèmes est peut-être due à un mauvais paramétrage de CPLEX.

Notons que la comparaison du temps de calcul entre l'implantation fondée sur CPLEX et les implantations fondées sur Choco n'a pas vraiment de sens : on cherche dans ce chapitre à comparer des différences d'approche du problème d'optimisation leximin, et non à comparer les outils de résolution eux-mêmes, comme nous l'avons fait remarquer en début de chapitre. Il serait en revanche intéressant de comparer dans le cadre de la programmation linéaire avec un outil comme CPLEX, l'algorithme fondé sur la contrainte **AtLeast** avec l'algorithme de [Dubois et Fortemps, 1999], l'algorithme inspiré de [Ehrgott, 2000], et l'algorithme de [Maschler *et al.*, 1992] fondé sur les transformations max-min.



Algorithme 6 fondé sur la contrainte **AtLeast**.

Algorithme 2, inspiré de [Dubois et Fortemps, 1999].

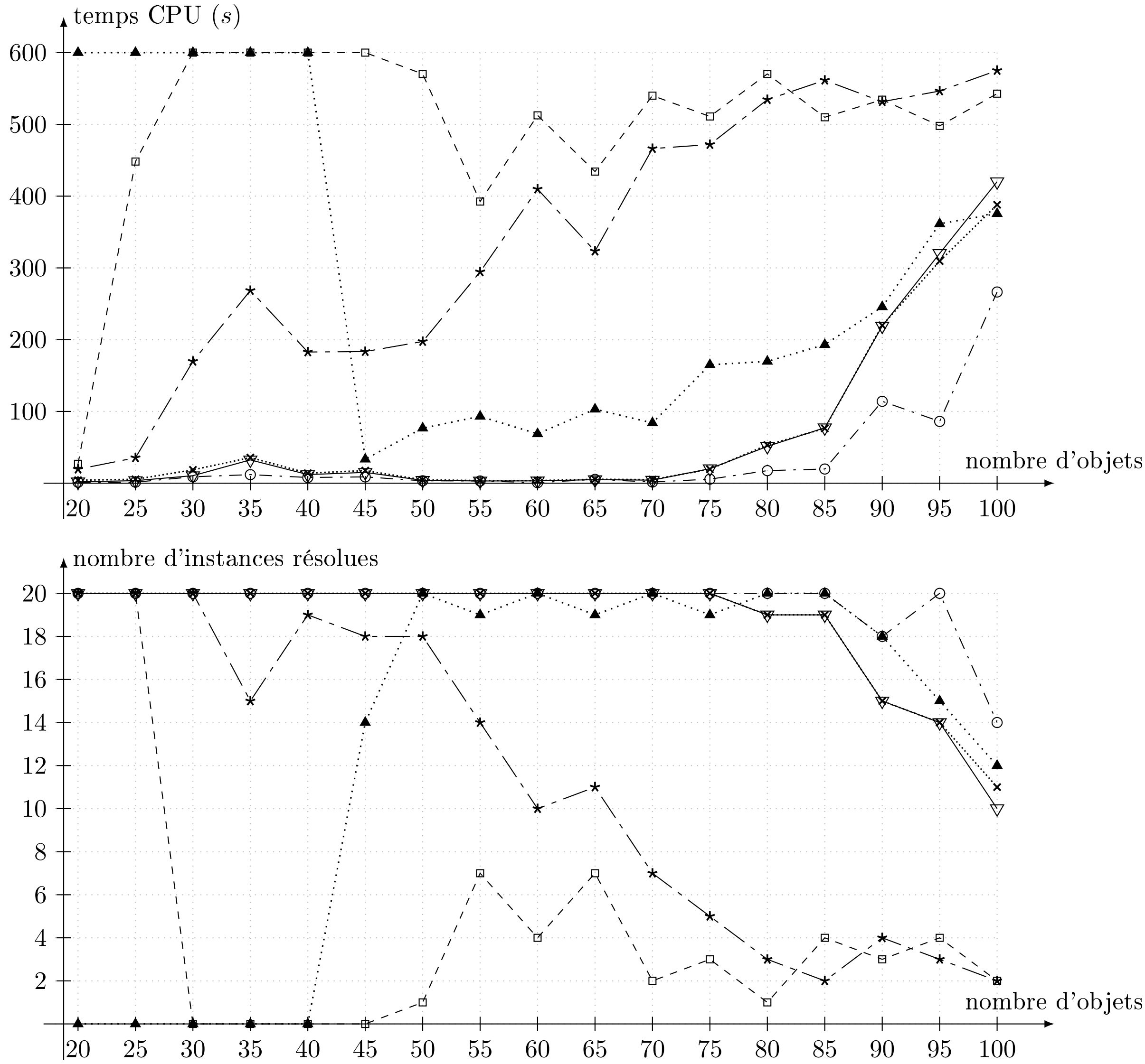
Algorithme fondé sur le calcul et la comparaison de toutes les solutions du problème, inspiré par [Ehrgott, 2000].

Algorithme 8 fondé sur les transformations max-min.

Algorithme 1 fondé sur la contrainte **Leximin**.

Algorithme 5 fondé sur la contrainte **Sort**.

Figure 6.2 — Comparaison des temps de calcul des algorithmes de résolution du problème [MAXLEXIMINCSP] sur des instances du type Pléiades simplifié à 10 agents.



Algorithme 6 fondé sur la contrainte **AtLeast**.

Algorithme 2, inspiré de [Dubois et Fortemps, 1999].

Algorithme fondé sur le calcul et la comparaison de toutes les solutions du problème, inspiré par [Ehrgott, 2000].

Algorithme 8 fondé sur les transformations max-min.

Algorithme 1 fondé sur la contrainte **Leximin**.

Algorithme 5 fondé sur la contrainte **Sort**.

Figure 6.3 — Comparaison des temps de calcul des algorithmes de résolution du problème [MAXLEXIMINCSP] sur des instances du type Pléiades simplifié à 20 agents.

6.2 Enchères combinatoires équitables

6.2.1 Description et modélisation du problème

Comme nous avons pu le voir au cours des chapitres précédents, le domaine des enchères combinatoires a suscité un intérêt croissant de la communauté du choix social et de l'intelligence artificielle ces dernières années, comme en témoignent les nombreuses publications qui lui sont consacrées [Cramton *et al.*, 2006; Sandholm, 1999, 2002; Rothkopf *et al.*, 1998; Lehmann *et al.*, 1999] (cette liste n'étant bien entendu pas exhaustive). Rappelons que le problème central des enchères combinatoires est le *Winner Determination Problem*, dédié à l'attribution des lots d'objets aux agents de manière à maximiser le gain du commissaire-priseur, ce qui correspond exactement à la maximisation de l'utilité collective utilitariste classique (la somme) si l'on considère que l'utilité d'un agent correspond au prix total auquel il estime les lots qu'il reçoit.

Si la sémantique des enchères combinatoires semble exclure l'égalitarisme comme critère d'agrégation des préférences individuelles, en revanche il est possible de s'appuyer sur ce problème et sur les travaux qui le concernent (notamment sur les langages d'expression des préférences) pour définir un problème de partage équitable d'objets. Ce n'est donc plus un problème d'enchères à proprement parler (car on ne s'intéresse plus aux gains du commissaire-priseur), mais un problème de partage défini comme suit :

Problème 18: Enchères combinatoires équitables

INSTANCE :	Un ensemble de n agents $\mathscr{N}$, un ensemble d'objets $\mathscr{O}$ et un ensemble de mises $\mathscr{M}_i$ par agent i, exprimées dans l'un des langages de lots pour les enchères combinatoires introduits aux chapitre 3.
SOLUTION :	Un partage $\overrightarrow{\pi}$ respectant la contrainte de préemption et tel qu'il n'existe aucun partage $\overrightarrow{\pi}'$ (respectant aussi la contrainte de préemption) tel que $(u(\pi_1), \ldots, u(\pi_n)) \prec (u(\pi'_1), \ldots, u(\pi'_n))$.

Les fonctions d'utilité des agents ont été définies lors de l'introduction des langages de lots dans la section 3.2.5.

6.2.2 Génération des instances

Comme nous l'avons fait remarquer en introduction de ce chapitre, s'il existe dans le domaine du choix social en général assez peu de travaux sur la génération d'instances de problèmes, le domaine particulier des enchères combinatoires fait exception à cette règle. Il existe en effet un générateur d'instances «réalistes» pour le problème d'enchères combinatoires, dont l'utilisation commence à se répandre et qui est devenu une référence dans le domaine : CATS [Leyton-Brown *et al.*, 2000] (disponible en ligne à l'URL `http://www.cs.ubc.ca/~kevinlb/CATS/`). Ce logiciel est dédié à la génération de mises pour les enchères combinatoires. La sémantique de ces mises est celle du *Winner Determination Problem* avec le langage OR* (voir la section 3.2.5 du chapitre 3). Les lots correspondant à ces mises ne sont pas mutuellement exclusifs sauf s'ils ont des objets en commun, et l'introduction d'objets factices permet de simuler des mises XOR, et donc d'enrichir le langage en permettant d'exprimer des substituabilités (rappelons que le langage OR seul ne le peut pas).

Décrivons rapidement la manière dont CATS génère l'ensemble de mises. Afin d'être les plus réalistes possibles, les algorithmes de génération des instances s'inspirent de problèmes réels. Cinq types de problèmes sont proposés (pour des informations plus détaillées, on pourra consulter l'article [Leyton-Brown *et al.*, 2000]) :

▷ **Chemins dans l'espace :** De nombreux problèmes réels d'enchères impliquent une recherche de chemins dans un graphe (tournées de véhicules, allocation de bande passante dans des réseaux, droit à utiliser un réseau ferré,...). CATS commence par générer un graphe approximativement planaire dont les nœuds (représentant des villes par exemple) sont placés de manière aléatoire dans le plan. Puis plusieurs ensembles de mises XOR sont générés : pour chaque ensemble on tire aléatoirement deux villes, puis on place un ensemble de mises XOR correspondant à tous les chemins possibles entre ces deux villes (la valeur des mises est aléatoire et dépendante de la distance euclidienne entre les deux villes).

▷ **Proximité dans l'espace :** Dans un autre grand type de problèmes, la complémentarité entre des objets est engendrée par leur proximité spatiale (exemple de vente aux enchères de terrains agricoles, droits de forages dans certaines zones, etc.). CATS génère un graphe qui est en fait un graphe d'adjacence (entre des terrains par exemple) : partant d'une grille parfaite (chaque nœud est connecté à 4 de ses voisins), il autorise avec une certaine probabilité, que certains nœuds soient connectés en diagonale, et que certaines connexions soient «oubliées» (représentant ainsi des terrains non rectangulaires). Les mises sont ensuite générées de manière incrémentale et aléatoire, en ajoutant les objets un par un dans les lots, avec une plus forte probabilité d'ajouter un objet s'il est connecté à l'un des objets du lot.

▷ **Relations arbitraires :** De manière générale (pour les ventes d'objets par exemple), il n'y a pas de relation spécifique entre les objets comme précédemment, mais on observe une certaine régularité dans les lots (régularité qui fait par exemple qu'un agent sera plus enclin à demander un lot {télévision, lecteur de DVD} qu'un lot {télévision, paire de chaussettes}). La génération des mises utilise une clique (un objet par nœud), avec une probabilité associée à chaque paire d'objets, qui indique la tendance qu'auront ces objets à s'associer. Comme précédemment, les lots sont générés de manière incrémentale.

▷ **Association temporelle :** Dans certains problèmes réels d'enchères, l'association entre les objets est temporelle. Par exemple, dans le problème de l'allocation de créneaux de décollage et d'atterrissage dans les aéroports (application 3), les créneaux sont liés par le fait qu'un avion qui effectue une liaison entre deux aéroports doit avoir les créneaux de décollage et d'atterrissage correspondant au temps de vol entre ces deux aéroports. Il y a aussi des lots substituables, correspondant aux différents couples décollage / atterrissage pour un même avion. La génération des lots prend en paramètre une carte des aéroports, et pour chaque vol, considère une utilité max u_{max} si le vol a ses créneaux décollage / atterrissage optimaux ; et une utilité inversement proportionnelle au retard engendré si ce n'est pas le cas.

▷ **Planification temporelle :** Il s'agit d'une formulation enchères combinatoires du problème de *job-shop scheduling*. Une usine lance une enchère pour les tranches d'utilisation d'une certaine ressource. Chaque enchérisseur a une tâche qui requiert un certain temps d'utilisation de la machine, et une *deadline*. Certaines tâches ont des *deadlines* additionnelles qui sont moins désirables pour l'enchérisseur. Dans la formulation orientée enchères combinatoires, un bien est un créneau d'utilisation de la machine. Les biens substituables correspondent aux différentes planifications possibles pour la même tâche.

En plus de ces cinq types de distribution, CATS peut générer les distributions classiques utilisées dans les papiers dédiés aux enchères combinatoires (distribution appelées *legacy*).

CATS considère que chaque agent exprime ses préférences sous forme de mises mutuellement exclusives (mises XOR) ; en d'autres termes, une seule mise peut être sélectionnée pour chaque agent. Puisque les mises XOR sont simulées par l'introduction d'objets factices (dans le langage OR*), on peut accéder à l'identité de l'agent concerné par une mise donnée grâce à l'objet factice qui accompagne le lot de la mise[1]. Puisque dans notre problème de recherche d'une solution lexi-

[1] Contrairement au *Winner Determination Problem* pour lequel l'information d'identité des agents n'est pas utile,

min optimale le nombre d'agents en jeu est un paramètre crucial, nous avons modifié légèrement l'exploitation du générateur afin de permettre à l'utilisateur de choisir un nombre d'agents inférieur au nombre d'agents généré par CATS, en fusionnant les mises de plusieurs agents différents de la manière suivante : nous groupons les agents deux par deux de manière aléatoire jusqu'à obtenir le nombre d'agents requis. Deux agents sont fusionnés simplement en groupant leurs préférences respectives en une mise OR* unique. Notons que cette manière de fusionner les mises n'est pas réellement satisfaisante, car le rapprochement des préférences des agents est complètement artificiel et crée des préférences non réalistes.

6.2.3 Résultats

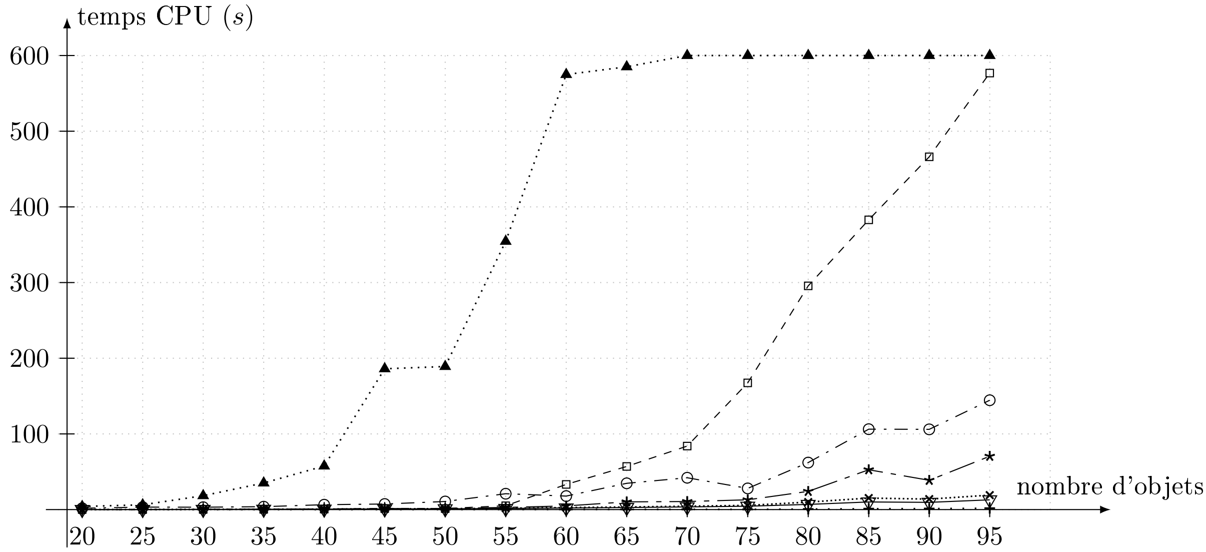
Nous avons mené plusieurs séries d'expérimentations sur les enchères combinatoires, afin toujours de mettre en valeur l'influence du nombre d'objets et du nombre d'agents sur les algorithmes. Les résultats de l'une de ces séries sont présentés en figure 6.4 (dans cette figure, nous avons omis le nombre d'instances résolues, car celui-ci est presque toujours égal à 20, sauf pour les deux algorithmes les moins performants). Pour cette série, les instances ont été générées à l'aide de CATS paramétré avec des lots de type «régions» (proximité dans l'espace), un nombre de mises égal au nombre d'objets et 20 agents. La figure met clairement en avant une nette infériorité des approches fondées sur les sous-ensembles saturés et sur la comparaison exhaustive de toutes les solutions. La taille des lots associée aux contraintes d'exclusion mutuelle pesant sur les lots peut expliquer l'échec du premier de ces deux algorithmes : dans de telles instances, il est très difficile de satisfaire tout le monde, donc il y a beaucoup d'agents non satisfaits dans les solutions leximin-optimales : on retombe dans le problème de calcul des sous-ensembles saturés.

L'algorithme fondé sur la contrainte **Leximin** semble beaucoup moins efficace que les algorithmes itératifs dans ce cas particulier (alors qu'il l'était beaucoup plus dans les instances de type Pléiades simplifié). L'explication que nous proposons est que le calcul des premières composantes du vecteur leximin-optimal dans les algorithmes itératifs résulte en un filtrage très important dès le début de la résolution (à cause de la morphologie du problème), ce qui peut expliquer leur efficacité dans ce cas bien précis.

La figure 6.5 présente des résultats relativement similaires à la figure 6.4. Pour cette série, les instances ont été générées avec des lots de type «arbitraires», un nombre de mises égal à 10 fois le nombre d'agents et 100 objets. L'échec de l'algorithme fondé sur les sous-ensembles saturés s'explique une nouvelle fois par l'augmentation du nombre d'agents par rapport au nombre d'objets qui reste fixe. On note un recul de l'efficacité de l'algorithme fondé sur les transformations max-min lorsque le nombre d'agents augmente : ces transformations sont certainement relativement coûteuses, car elles nécessitent l'introduction de variables supplémentaires dont le nombre croît de manière quadratique avec le nombre d'agents. Les algorithmes les plus efficaces sont ceux qui sont fondés sur la contrainte **Sort** et sur la contrainte **AtLeast**.

Enfin, nous avons comparé les temps de calcul d'une solution leximin-optimale au temps de calcul d'une solution somme-optimale, pour le même modèle, en programmation par contraintes. Bien entendu, la programmation par contraintes n'est pas l'outil le plus adapté pour traiter le *Winner Determination Problem*, et en particulier les solveurs *ad-hoc* proposés dans la littérature sur les enchères combinatoires sont bien plus performants. Nous pouvons toutefois noter que le calcul d'une solution leximin-optimale n'est pas énormément plus coûteuse que le calcul d'une solution somme-optimale dans le même modèle.

nous avons besoin de connaître cette identité pour le problème de recherche d'une solution leximin-optimale.

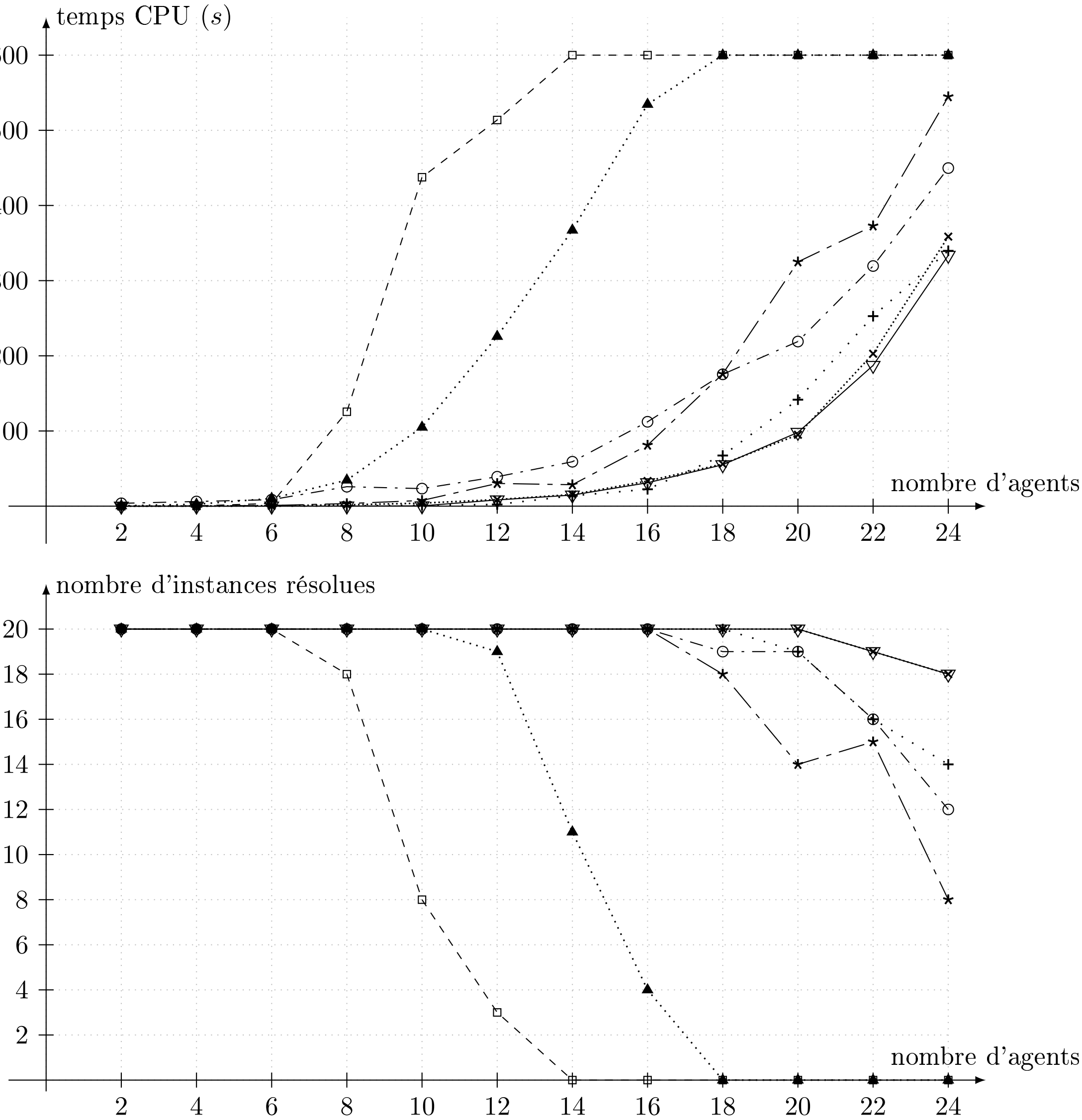


- Algorithme 6 fondé sur la contrainte **AtLeast**.
- Algorithme 2, inspiré de [Dubois et Fortemps, 1999].
- Algorithme fondé sur le calcul et la comparaison de toutes les solutions du problème, inspiré par [Ehrgott, 2000].
- Algorithme 8 fondé sur les transformations max-min.
- Algorithme 1 fondé sur la contrainte **Leximin**.
- Algorithme 5 fondé sur la contrainte **Sort**.
- Calcul de la solution optimale au sens de la somme (correspond au *Winner Determination Problem*).

Figure 6.4 — Comparaison des temps de calcul des algorithmes de résolution du problème [MAXLEXIMINCSP] sur des instances de type enchères combinatoires à 20 agents.

Heuristique de choix des variables Tout comme pour les instances de type Pléiades simplifié, nous avons décliné l'heuristique de choix des variables suggérée dans la section 5.4.4 aux problèmes d'enchères combinatoires. Plus précisément, l'heuristique est la suivante : on choisit comme prochain lot à allouer le premier d'entre les lots non encore alloués, selon l'ordre lexicographique sur les critères suivants : (1) l'utilité courante de l'agent qui a placé cette mise, (2) le prix du lot. En d'autres termes, le prochain lot à allouer est celui qui a le prix le plus fort parmi ceux de l'agent le moins satisfait. Encore une fois, on relève une amélioration notable des performances grâce à l'utilisation de cette heuristique.

Équité de la solution Intéressons-nous brièvement à la caractérisation de l'équité des solutions obtenues par maximisation de l'ordre social leximin par rapport à l'équité des solutions obtenues par résolution du *Winner Determination Problem* avec critère utilitariste classique. Nous avons tracé sur la figure 6.6 les courbes de Lorenz des deux solutions utilitariste classique et égalitariste leximin pour une instance du problème d'enchères combinatoires avec 20 agents. La représentation graphique de ces deux courbes est relativement démonstrative : la solution égalitariste est beaucoup plus proche de la droite d'égalité parfaite entre les agents. En terme d'indices d'inégalité, le calcul des indices de Gini pour les deux profils donne une valeur d'environ 0,83 pour l'utilitarisme classique et 0,64 pour la solution leximin. Rappelons que plus l'indice d'inégalité est proche de 0, plus la distribution



Algorithme 6 fondé sur la contrainte **AtLeast**.

Algorithme 2, inspiré de [Dubois et Fortemps, 1999].

Algorithme fondé sur le calcul et la comparaison de toutes les solutions du problème, inspiré par [Ehrgott, 2000].

Algorithme 8 fondé sur les transformations max-min.

Algorithme 1 fondé sur la contrainte **Leximin**.

Algorithme 5 fondé sur la contrainte **Sort**.

Calcul de la solution optimale au sens de la somme (correspond au *Winner Determination Problem*).

Figure 6.5 — Comparaison des temps de calcul des algorithmes de résolution du problème [MAXLEXIMINCSP] sur des instances de type enchères combinatoires à 100 objets.

est égalitaire.

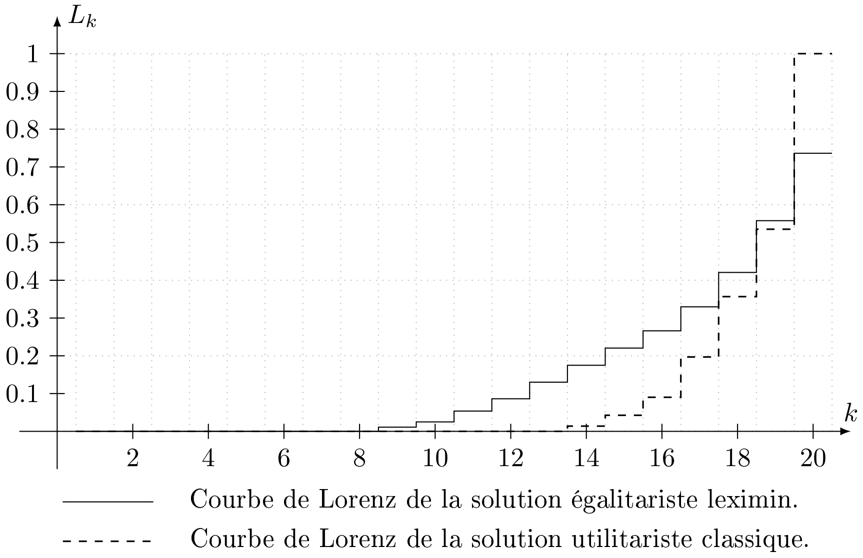


Figure 6.6 — Courbes de Lorenz des solutions utilitariste classique et égalitariste leximin pour une instance du problème d'enchères combinatoires.

6.3 Problème de partage de biens indivisibles générique

6.3.1 Description du modèle

Nous avons implanté le modèle du problème de partage de biens indivisibles générique fondé sur la représentation des préférences sous forme de logique pondérée et la représentation des contraintes sous forme de logique, problème que nous avons introduit dans la définition 3.38 dans le chapitre 3. Rappelons que ce problème est fondé sur :

- ▷ un ensemble d'agents $\mathscr{N}$;
- ▷ un ensemble d'objets $\mathscr{O}$;
- ▷ un ensemble de contraintes $\mathscr{C}$ exprimées sous forme logique sur le langage $L_{\mathscr{O}}^{alloc}$;
- ▷ les préférences des agents exprimées par un ensemble de formules logiques pondérées sur $\mathcal{L}_{\mathscr{O}}$;
- ▷ deux opérateurs d'agrégation.

Dans notre implantation de ce modèle, nous avons ajouté la possibilité d'exprimer deux types de contraintes spécifiques en plus des contraintes logiques :

- ▷ des contraintes de préemption, portant sur des ensembles d'objets donnés ;
- ▷ des contraintes de volume, définie par un volume maximum et portant sur des ensembles d'objets donnés auxquels on attribue un volume.

Ces contraintes étant omniprésentes dans les problèmes de partage réels, leur introduction spécifique permet de simplifier leur expression et leur prise en compte par les algorithmes de résolution, même si, comme nous l'avons précisé lors de l'introduction de ce modèle au chapitre 3, ces contraintes peuvent être encodées par des formules logiques de notre langage.

Dans notre implantation, les contraintes logiques peuvent être spécifiées de différentes manières :

- ▷ comme une formule logique générique (spécifiée par application récursive d'un opérateur logique sur des sous-formules) ;

▷ comme une clause du langage propositionnel ;
▷ comme une formule en forme normale disjonctive (par spécification des cubes sous forme d'un tableau).

De même, les formules logiques associées aux préférences peuvent être définies de différentes manières :

▷ comme une formule logique générique ;
▷ comme une formule en forme normale conjonctive ;
▷ comme une formule en forme normale disjonctive.

La possibilité de définir les contraintes et les préférences de différentes manières permet de prendre en compte de manière spécifique ces types de formules spécifiques (par exemple en introduisant des mécanismes de filtrage dédiés aux clauses ou aux cubes), afin d'une part de faciliter la spécification des problèmes, et d'autre part d'améliorer l'efficacité des algorithmes de résolution qui leur sont dédiés.

6.3.2 Génération des instances

Nous avons développé un générateur d'instances de ce problème, dont l'objectif est, tout comme le logiciel CATS pour les enchères combinatoires, de créer des instances réalistes de ce problème. Ce générateur, disponible en ligne à l'adresse `http://www.cert.fr/dcsd/THESES/sbouveret/benchmark2007/`, permet de créer trois types d'instances : générique, Pléiades et Spot.

6.3.2.1 Instances de type générique

Ce type d'instances est destiné à représenter des problèmes de partage génériques relativement simples. Les caractéristiques de ces instances sont les suivantes :

▷ les préférences sont constituées de formules en forme normale disjonctive (partant du principe qu'un agent sera plus enclin à exprimer naturellement ce genre de préférences) de taille et de poids aléatoires ;
▷ les contraintes logiques sont de deux types :
  - interdiction d'un objet particulier à un agent particulier,
  - interdiction d'attribuer un ensemble donné d'objets de taille aléatoire au même agent.
▷ la contrainte de préemption portant sur l'ensemble des objets est présente ou non.

Le générateur est bien entendu entièrement paramétrable. La manière de le paramétrer et les valeurs par défaut sont spécifiés sur la page web dédiée au générateur d'instances.

6.3.2.2 Instances de type Pléiades

Ce deuxième type d'instances du problème de partage de biens indivisibles est destiné à introduire un modèle du problème de partage de la constellation de satellite Pléiades qui soit légèrement plus réaliste que le problème linéaire introduit dans la section 6.1 de ce chapitre. La génération des instances se fait en deux temps.

Dans un premier temps, le programme génère une instance du modèle fondé sur les éléments suivants :

▷ La constellation est constituée d'un seul satellite, et le problème considère n_r révolutions du satellite autour de la Terre.
▷ Les agents fournissent un ensemble de demandes d'observation.
▷ Une demande d'observation est constituée d'une (requête simple) ou de plusieurs (requêtes *stereo* ou de zones larges) acquisitions qui doivent toutes être satisfaites pour satisfaire la

demande. Un poids est aussi associé à la demande, poids reflétant l'importance de cette demande pour l'agent.

- Une acquisition est constituée de plusieurs opportunités : une par révolution. Une acquisition est satisfaite si et seulement si au moins l'une des opportunités est satisfaite. Chaque acquisition est caractérisée par un numéro d'identification et le numéro de la révolution qu'elle concerne.
- Un ensemble de contraintes de volume simule les contraintes physiques, tout comme dans le premier modèle simplifié du problème.
- Certaines demandes d'observation sont de type particulier : les «demandes communes». Ces demandes d'observation concernent tous les agents en même temps, et si elles sont satisfaites, tous les agents sont satisfaits.
- Certaines demandes d'observation complexes comme les demandes *stereo* ou les images concernant une zone large doivent être acquises depuis la même révolution.

L'instance de ce modèle créée par le générateur est ensuite transformée en une instance du problème de partage de biens indivisibles. Ce générateur (entièrement paramétrable) permet de créer des instances légèrement plus réalistes que celles du problème Pléiades simplifié introduit en début de chapitre.

6.3.2.3 Instances de type Spot

Le troisième type d'instances particulières considéré pour le problème de partage de biens indivisibles est assez similaire aux instances de type Pléiades. Il est fondé sur le modèle constitué des éléments suivants :

- La constellation est formée de plusieurs satellites, et le problème considère n_r révolutions des satellites autour de la Terre. Les satellites possèdent chacun 3 instruments (avant, milieu et arrière), permettant de multiplier les opportunités d'acquisition. Les satellites ne sont pas agiles, mais un système de miroirs permet de changer dans une certaine mesure l'angle de visée latéral des instruments.
- Les agents fournissent un ensemble de demandes d'observation.
- Une demande observation est constituée d'une (requête simple) ou de plusieurs (requêtes *stereo*) acquisitions qui doivent toutes être satisfaites pour satisfaire la demande. Un poids est aussi associé à la demande, poids reflétant l'importance de cette demande pour l'agent.
- Une acquisition est constituée de plusieurs opportunités : une par révolution, par instrument et par satellite (sauf pour les images *stereo*, qui doivent être impérativement acquises par les instruments avant et arrière, lors de la même orbite). Chaque opportunité est caractérisée par l'instrument et la révolution concernés, un angle de déportation, et des temps de début et de fin de prise de vue. Les temps sont générés de manière aléatoire, soit selon une loi uniforme, soit concentrés autour de certains «points chauds», représentant des points d'intérêt particulier (comme des zones de conflits par exemple).
- Les contraintes sont de deux types :
  - des contraintes de volume, simulant les contraintes de consommation à bord : à chaque acquisition est associée une consommation de ressources qui dépend de sa durée, et chaque satellite ne peut acquérir plus d'images que sa consommation limite ne le permet ;
  - des contraintes d'exclusion mutuelle, fondées sur les temps de début et de fin des opportunités et sur leurs angles de déportation : le générateur effectue un calcul de faisabilité d'enchaînement de deux opportunités concernant le même satellite, et ajoute une contrainte d'exclusion mutuelle si cet enchaînement est impossible.

Après avoir créé une instance de ce problème particulier, le générateur la transforme, tout comme précédemment, en une instance du problème de partage de biens indivisibles.

6.3.3 Résultats

Les expérimentations des algorithmes sur ce type d'instances fournissent des résultats relativement décevants : les algorithmes sont rapidement mis en échec sur un très petit nombre d'objets et un nombre d'agents raisonnable, comme nous pouvons le voir sur la figure 6.7 et sur le tableau 6.1, concernant respectivement des instances de type logique générique et des instances de type Pléiades logique. Cette relative inefficacité peut s'expliquer par la difficulté des instances, qui impliquent un grand nombre de contraintes logiques[2]. On pourra néanmoins remarquer la supériorité relative des approches itératives (à l'exception de celle qui est fondée sur les transformations max-min) sur les approches fondées sur la contrainte **Leximin** et sur la comparaison exhaustive de toutes les solutions. On pourra remarquer aussi comme toujours la proximité entre les approches fondées sur la contrainte **AtLeast** et sur la contrainte **Sort**.

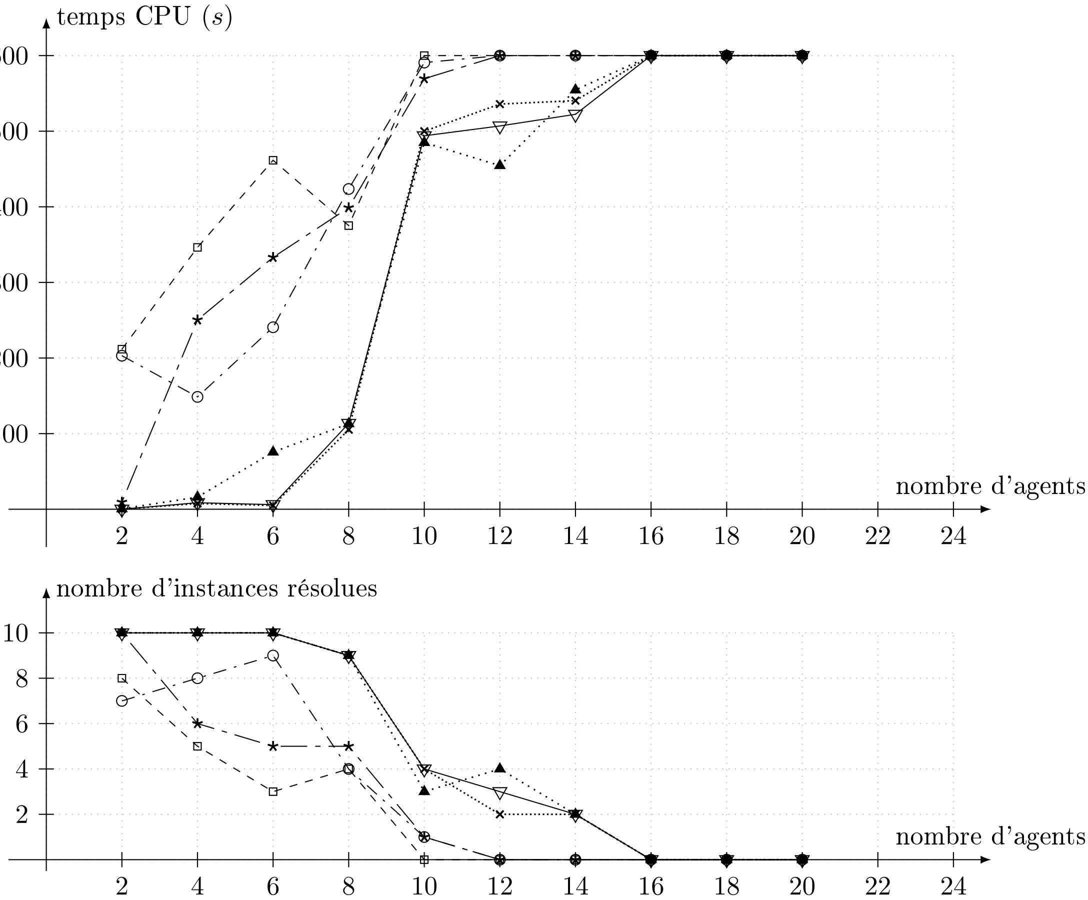
Instance		**AtLeast**				sous-ensembles saturés				comparaison exhaustive			
n	p	min	max	moy	nb	min	max	moy	nb	min	max	moy	nb
2	20	4	309	**45**	10	281	639	447	10	6	1675	295	10
2	25	6	942	202	10	369	1831	705	10	4	97940	12924	10
2	30	10	1448	275	10	407	2060	867	10	23	15770	4284	10
6	20	15	(—)	240070	6	2564	(—)	241853	6	38024	(—)	494419	2
6	25	98	(—)	364452	4	2887	(—)	371697	4	4216	(—)	540421	1
6	30	75	(—)	**423021**	3	3040	(—)	426813	3	(—)	(—)	(—)	0
10	20	231	(—)	**540023**	1	63931	(—)	546393	1	(—)	(—)	(—)	0

Instance		Max-min				**Leximin**				**Sort**			
n	p	min	max	moy	nb	min	max	moy	nb	min	max	moy	nb
2	20	3	552	87	10	4	56844	5968	10	3	76	21	10
2	25	4	3863	942	10	2	18310	5330	10	4	959	**192**	10
2	30	9	13299	1517	10	21	8223	2777	10	10	1371	**266**	10
6	20	21	(—)	360056	4	94769	(—)	513778	2	16	(—)	**240042**	6
6	25	236	(—)	540023	1	5037	(—)	540503	1	97	(—)	**364449**	4
6	30	(—)	(—)	(—)	0	(—)	(—)	(—)	0	85	(—)	424616	3
10	20	267	(—)	540026	1	(—)	(—)	(—)	0	240	(—)	540024	1

Tableau 6.1 — Temps de calcul (en ms) des algorithmes de résolution du problème [MAXLEXIMINCSP] sur des instances de type problème logique Pléiades, à n agents et p objets.

Heuristique de choix des variables L'heuristique de choix des variables est plus difficile à mettre en œuvre dans le contexte des instances du problème logique générique, car l'effet de l'attribution d'un objet sur la satisfaction d'un agent n'est pas direct, du fait de la complexité sémantique des préférences. En outre, des considérations telles que le nombre de demandes dans lesquels apparaît l'objet et leur longueur doivent être prises en compte pour le choix des variables. Nous ne nous sommes pas encore penchés sur ce problème ; l'heuristique utilisée actuellement est la suivante : on choisit comme prochain objet à attribuer un objet apparaissant dans la demande de plus fort poids de l'agent le moins satisfait.

[2]Les paramètres par défaut pour le générateur aléatoire générique sont les suivants : entre $n/2$ et $3n/2$ contraintes particulières, entre 0 et $p/2$ contraintes d'exclusion concernant entre 2 et $p/2$ objets, entre 1 et 10 formules par agent concernant entre 1 et 15 objets chacune.



	Algorithme 6 fondé sur la contrainte **AtLeast**.
	Algorithme 2, inspiré de [Dubois et Fortemps, 1999].
	Algorithme fondé sur le calcul et la comparaison de toutes les solutions du problème, inspiré par [Ehrgott, 2000].
	Algorithme 8 fondé sur les transformations max-min.
	Algorithme 1 fondé sur la contrainte **Leximin**.
	Algorithme 5 fondé sur la contrainte **Sort**.

Figure 6.7 — Comparaison des temps de calcul des algorithmes de résolution du problème [MAXLEXIMINCSP] sur des instances de type problème logique générique, à 20 objets.

6.4 Problème d'affectation de sujets de travaux expérimentaux

Le problème d'affectation de TRavaux EXpérimentaux (« TREX ») est un exemple typique de problèmes de distribution de ressources indivisibles préemptives (le même objet ne peut être donné à plusieurs agents) sous contraintes, et avec un ensemble de préférences, tels qu'on peut en trouver dans les institutions comme des écoles, des universités, ou encore des entreprises. Le but de ce genre de problèmes est d'affecter un ou plusieurs sujets, tâches, ou objets, à chacun des agents présents, en tenant compte de leurs préférences exprimées comme un ordre sur les objets. Ce problème est présenté succinctement dans le chapitre d'introduction (application 4).

Nous avons à disposition une instance réelle, provenant de l'expression des choix des élèves de deuxième année à SUPAÉRO et des contraintes spécifiques aux sujets en 2005–2006. Nous allons dans un premier temps décrire l'instance particulière à laquelle nous avons affaire, puis nous nous intéresserons au calcul d'une solution leximin-optimale, et nous montrerons que toutes les approches testées ont échoué sur ce problème.

6.4.1 Description de l'instance réelle

On doit affecter un ensemble de 38 sujets de TREX à un ensemble d'élèves, regroupés en 63 binômes, formant eux-mêmes deux groupes D et H de 31 et 32 binômes respectivement. Les sujets sont regroupés en 9 différents pôles de compétence (Mathématiques, Informatique, Aérodynamique, ...) de la manière suivante : $\langle [\![1,5]\!], [\![6,10]\!], [\![11,13]\!], [\![14,15]\!], [\![16,18]\!], [\![19,22]\!], [\![23,30]\!], [\![31,37]\!], [\![38]\!] \rangle$.

On doit allouer les TREX aux binômes, sous les contraintes suivantes.

- ▷ **Préemption :** Un même sujet ne peut être affecté à deux binômes différents du même groupe dans la même série.
- ▷ **Fourniture :** Chaque binôme doit être pourvu d'un TREX exactement pour chaque série (1 et 2).
- ▷ **Variété de sujets :** Un même binôme ne peut se voir affecter deux sujets du même pôle de compétences.
- ▷ **Sujets spécifiques :** Certains sujets ont des contraintes spécifiques :
  - les 6 sujets suivants ne peuvent pas être choisis par un binôme du groupe D en série 2 : 5, 9, 32, 34, 35, 36 ;
  - les 6 sujets suivants ne peuvent pas être choisis par un binôme du groupe D en série 1 : 8, 18, 29, 31, 33, 37 ;
  - le sujet 38 ne peut être choisi par aucun binôme du groupe D, que ce soit en série 1 ou en série 2 ;
  - les 6 sujets suivants ne peuvent pas être choisis par un binôme du groupe H en série 2 : 8, 29, 31, 33, 37, 38 ;
  - les 6 sujets suivants ne peuvent pas être choisis par un binôme du groupe H en série 1 : 5, 18, 32, 34, 35, 36 ;

Chaque binôme a exprimé une préférence sur l'ensemble des sujets, sous la forme d'un ordre total (pas d'*ex-aequo* possible), en attribuant au sujet préféré la valeur 1, et au sujet le plus bas dans l'ordre de préférences la valeur 38. Le manière de calculer l'utilité d'un binôme en fonction des deux sujets qu'il reçoit n'est pas spécifiée dans les données du problème. Il y a toutefois trois manières intuitives de définir cette utilité :

- ▷ $u_i = 76 - k_1 - k_2$, si k_1 (resp. k_2) est la place du sujet reçu par le binôme i en série 1 (resp. série 2), en d'autres termes, la désutilité d'un agent à la somme (ou à la moyenne) des rangs des deux sujets qu'il reçoit ;
- ▷ $u_i = 38 - \max(k_1, k_2)$ (seul compte le pire des sujets), avec variante leximax possible ;

▷ $u_i = 38 - \min(k_1, k_2)$ (seul compte le meilleur des sujets), avec variante leximin possible.
Ces trois formulations de l'utilité individuelle sont valables d'un point de vue sémantique. Nous penchons plutôt pour la formulation ne prenant en compte que le pire des sujets, pour une raison d'efficacité de résolution : ce choix permet de modéliser le problème d'optimisation max-min «presque» comme un problème de flot dans un graphe, comme nous allons le voir plus loin.

L'objectif du problème est de trouver une affectation des sujets aux binômes qui satisfait toutes les contraintes, et soit équitable et efficace. Bien entendu, ce critère est assez flou et laisse toute latitude à l'interprétation des notions d'équité et d'efficacité. Toutefois, la modélisation de ces notions par le biais du préordre leximin nous semble relativement pertinent ici. Notons que le critère qui actuellement utilisé est le critère max-min / max-sum, dont nous avons parlé au tout début du chapitre 5.

Nous pouvons remarquer qu'étant donnée la spécification du problème, celui-ci peut être séparé en deux sous-problèmes disjoints : celui du groupe D et celui du groupe H. Dans les sous-sections suivantes consacrées à la résolution du problème, nous ne nous intéresserons qu'à la première moitié du problème, c'est-à-dire celle qui concerne le groupe D (sauf dans la formulation mathématique qui implique de manière explicite les deux groupes).

6.4.2 Modélisation et résolution du problème

6.4.2.1 Formulation mathématique du problème

Modélisation Ce problème peut être modélisé de manière naturelle sous la forme d'un réseau de contraintes constitué d'un ensemble de variables 0–1, correspondant à l'affectation des sujets aux binômes de chaque série. Ces variables sont regroupées en quatre matrices correspondant aux quatre couples possibles (groupe, série). Le réseau de contraintes est donc le suivant :

▷ **variables :** 4 matrices $\mathbf{D^{(1)}}$, $\mathbf{D^{(2)}}$ (de tailles 31×38), $\mathbf{H^{(1)}}$, $\mathbf{H^{(2)}}$ (de tailles 32×38) de variables ; pour une matrice X, on notera $x_{i,j}$ ses éléments ;

$$\begin{array}{r|c|c}
 & \overbrace{1 \quad \cdots \quad 38}^{\text{Série 1}} & \overbrace{1 \quad \cdots \quad 38}^{\text{Série 2}} \\
\hline
\text{Groupe D} \left\{ \begin{array}{c} 1 \\ \vdots \\ 31 \end{array} \right. & \mathbf{D^{(1)}} & \mathbf{D^{(2)}} \\
\hline
\text{Groupe H} \left\{ \begin{array}{c} 1 \\ \vdots \\ 32 \end{array} \right. & \mathbf{H^{(1)}} & \mathbf{H^{(2)}}
\end{array}$$

▷ **domaines :** tous les domaines sont $\{0, 1\}$;

▷ **contraintes :**

$$\left.\begin{array}{l}
\bullet \ \forall j \in [\![1, 38]\!],\ \sum_{i=1}^{31} \mathbf{d_{ij}^{(1)}} \leq 1 \text{ et } \sum_{i=1}^{31} \mathbf{d_{ij}^{(2)}} \leq 1, \\
\bullet \ \forall j \in [\![1, 38]\!],\ \sum_{i=1}^{32} \mathbf{h_{ij}^{(1)}} \leq 1 \text{ et } \sum_{i=1}^{32} \mathbf{h_{ij}^{(2)}} \leq 1,
\end{array}\right\} \text{contraintes de préemption}$$

$$\left.\begin{array}{l}
\bullet \ \forall i \in [\![1, 31]\!],\ \sum_{j=1}^{38} \mathbf{d_{ij}^{(1)}} = 1 \text{ et } \sum_{j=1}^{38} \mathbf{d_{ij}^{(2)}} = 1, \\
\bullet \ \forall i \in [\![1, 32]\!],\ \sum_{j=1}^{38} \mathbf{h_{ij}^{(1)}} = 1 \text{ et } \sum_{j=1}^{38} \mathbf{h_{ij}^{(2)}} = 1,
\end{array}\right\} \text{contraintes de fourniture}$$

$$\left.\begin{array}{l}
\bullet\ \forall i \in [\![1,31]\!],\ \sum_{j=1}^{5} \mathbf{d_{i,j}^{(1)}} + \mathbf{d_{i,j}^{(2)}} \leq 1,\\
\bullet\ \forall i \in [\![1,32]\!],\ \sum_{j=1}^{5} \mathbf{h_{i,j}^{(1)}} + \mathbf{h_{i,j}^{(2)}} \leq 1,\\
\quad \ldots\\
\bullet\ \forall i \in [\![1,31]\!],\ \sum_{j=31}^{37} \mathbf{d_{i,j}^{(1)}} + \mathbf{d_{i,j}^{(2)}} \leq 1,\\
\bullet\ \forall i \in [\![1,32]\!],\ \sum_{j=31}^{37} \mathbf{h_{i,j}^{(1)}} + \mathbf{h_{i,j}^{(2)}} \leq 1,
\end{array}\right\}\ \text{contraintes de variété}$$

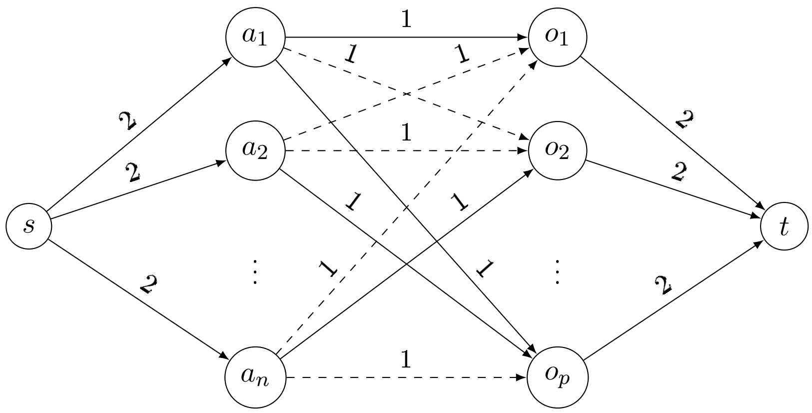
$$\left.\begin{array}{l}
\bullet\ \forall j \in \{5,9,32,34,35,36\},\ \sum_{i=1}^{31} \mathbf{d_{i,j}^{(1)}} = 1 \text{ et } \sum_{i=1}^{31} \mathbf{d_{i,j}^{(2)}} = 0,\\
\bullet\ \forall j \in \{8,18,29,31,33,37\},\ \sum_{i=1}^{32} \mathbf{d_{i,j}^{(1)}} = 0 \text{ et } \sum_{i=1}^{32} \mathbf{d_{i,j}^{(2)}} = 1,\\
\bullet\ \sum_{i=1}^{31} \mathbf{d_{i,38}^{(1)}} + \sum_{i=1}^{31} \mathbf{d_{i,38}^{(2)}} = 0,\\
\bullet\ \forall j \in \{8,29,31,33,37,38\},\ \sum_{i=1}^{31} \mathbf{h_{i,j}^{(1)}} = 1 \text{ et } \sum_{i=1}^{31} \mathbf{h_{i,j}^{(2)}} = 0,\\
\bullet\ \forall j \in \{5,18,32,34,35,36\},\ \sum_{i=1}^{32} \mathbf{h_{i,j}^{(1)}} = 0 \text{ et } \sum_{i=1}^{32} \mathbf{h_{i,j}^{(2)}} = 1
\end{array}\right\}\ \text{contraintes spécifiques}$$

Résolution Nous avons implanté en Java deux versions du modèle précédent : une version programmation par contraintes, destinée à être traitée avec la bibliothèque Choco, et une version en programmation linéaire, dédiée à l'interface avec le programme ILOG CPLEX. Contre toute attente, les approches tentées s'avèrent complètement inefficace sur le problème d'optimisation max-min (préalable à toute optimisation leximin) pour cette instance réelle : Choco comme CPLEX échouent à trouver une solution en un temps raisonnable (plusieurs jours), et ce, avec n'importe laquelle des trois fonctions d'utilité individuelle citées ci-avant. Ce résultat est d'autant plus étonnant que le calcul du max-somme est très rapide (quelques secondes). Remarquons que la méthode actuelle utilisée à SUPAÉRO est fondée sur des optimisations max-somme successives avec contrainte de borne sur l'utilité minimum : à chaque pas on essaie d'augmenter la borne de l'utilité minimum, jusqu'à ce que le problème mette trop de temps à être résolu. Il s'agit d'une approximation du critère max-min / max-somme (dont nous avons contesté la sémantique au début du chapitre 5).

6.4.2.2 Problème de flot sous contraintes

La formulation mathématique du problème dissimule quelque peu sa structure réelle, héritée de tous les problèmes d'affectation «de type couplage». Ce genre de problèmes se modélisent de manière assez naturelle sous la forme de problèmes de couplages dans un graphe bipartite, ou encore de problèmes de flots (ce qui est équivalent), exactement de la même manière que dans la proposition 4.23 page 162, détaillant la complexité du problème de partage de biens indivisibles pour des demandes atomiques et une fonction d'utilité collective de type min. Si l'on s'intéresse au cas où la fonction d'utilité individuelle est telle que seul compte le pire des sujets, on peut «presque» modéliser le problème d'affectation sous la forme du problème de flot présenté dans la figure 6.8. Le problème d'optimisation max-min revient à chercher la valeur K minimale telle qu'il existe un flot de valeur $2n$ dans le graphe pour lequel on a enlevé toutes les arêtes (a_i, o_j) tels que le sujet o_j est classé en dessous du rang K dans les préférences du binôme a_i. Cette valeur de K peut être calculée par dichotomie sur tout l'ensemble de valeurs possibles.

Le seul obstacle à cette modélisation est l'introduction des contraintes spécifiques et des contraintes de variété, qui rendent donc impossible la réduction de ce problème en un problème de flot. Cependant, on peut se fonder sur l'expression du problème de flot sans les contraintes spécifiques et de variété pour dériver un nouveau modèle linéaire : les variables correspondent aux arêtes sélectionnées ou non dans le graphe, et les contraintes expriment la conservation du flot à chaque nœud et les contraintes de capacité des arcs. On ajoute ensuite dans le modèle linéaire les contraintes spécifiques et de variété du problème d'affectation de sujets, et on détermine la valeur max-min par calculs de «flots» successifs comme ci-dessus.



Les arcs en traits pleins (resp. pointillés) correspondent aux couples (a_i, o_j) tels que le sujet o_j est classé en dessous (resp. au-dessus) du rang K dans les préférences du binôme a_i.

Figure 6.8 — Problème de flot correspondant au problème d'affectation de sujets de TREX.

Aussi étonnant que cela puisse paraître, cette seconde modélisation s'avère beaucoup plus efficace que la première pour le calcul de la valeur max-min optimale. À titre d'exemple, CPLEX met environ 10 millisecondes pour calculer cette valeur, ce qui est une avancée remarquable dans la résolution du problème.

Cette approche pose cependant problème pour le calcul d'une solution leximin-optimale. En effet, le fait que le calcul de la valeur max-min optimale soit effectué par améliorations successives d'une borne qui n'est pas une variable du solveur linéaire nous empêche d'utiliser un algorithme de résolution tel que celui qui est fondé sur la contrainte **AtLeast** (algorithme 6, page 191). Parmi les approches du problème [MAXLEXIMINCSP] présentées au chapitre 5, seul l'algorithme 2 inspiré de [Dubois et Fortemps, 1999] peut s'adapter à cette modélisation. Cependant, l'utilisation de cet algorithme dans ce cas précis se heurte à l'écueil du calcul des sous-ensembles saturés, que nous avons décrit lors de l'introduction de cet algorithme, et que nous avons mis en évidence sur les expérimentations précédentes. Dans le problème d'affectation de sujets, on peut estimer que les sous-ensembles saturés de cardinalité minimale contiennent en moyenne 10 variables. Si chaque résolution du problème de flot dure en moyenne 10 millisecondes, le parcours de tous les sous-ensembles de variables objectif (il y en a 62 dans notre cas) de taille inférieure ou égale à 10 durera $3,65 \times 10^6$ heures, soit 478 années : nous tombons dans le piège combinatoire.

6.5 Conclusion

Nous avons présenté dans ce chapitre un comparatif de l'efficacité des algorithmes introduits au chapitre 5 sur quatre types de jeux de tests : Pléiades simplifié, enchères combinatoires, partage de biens indivisibles à base de logique, et affectation de sujets de travaux expérimentaux. Ces tests ont permis de mettre en évidence la relative efficacité des approches fondées sur la contrainte **AtLeast** et la contrainte **Sort**, qui fournissent des temps de calcul corrects sur la plupart des instances des deux premiers jeux de tests. L'algorithme fondé sur la contrainte **Leximin** se montre plus efficace que les autres sur les instances Pléiades linéaires, mais moins efficace sur les autres types d'instances. L'algorithme inspiré de [Dubois et Fortemps, 1999] se révèle sans surprise à éviter à tout prix sur des instances produisant de nombreux *ex-aequo*, mais fournit des temps de calcul tout-à-fait raisonnables dès que l'on sort de ce type d'instances. L'algorithme fondé sur les transformations max-min a une efficacité limitée, et enfin l'algorithme fondé sur une comparaison exhaustive de toutes les solutions est à éviter.

Cependant, comme le montrent les résultats obtenus sur les instances des deux derniers jeux de test (et en particulier sur l'instance réelle du problème d'affectation de travaux expérimentaux), les algorithmes se révèlent complètement inefficaces sur des instances plus complexes ou sur un problème réel d'affectation. Cette constatation plaide donc en faveur d'autres approches, telles que les algorithmes de type k-meilleurs [Gonzales *et al.*, 2006] par exemple. Le développement d'algorithmes de recherche incomplets, du type recherche locale, peut constituer une autre alternative crédible et en tout cas prometteuse.

Conclusion

Synthèse des contributions

L'expression et la prise en compte de contraintes et de demandes complexes dans les problèmes de partage de biens indivisibles font émerger de nouvelles problématiques d'importance, liées notamment à la représentation compacte des préférences et des contraintes, à la complexité théorique des problèmes de partage, et aux aspects algorithmiques liés au calcul d'un partage optimal. Jusqu'ici, la plupart des travaux sur le partage de biens indivisibles se sont concentrés :

- ▷ d'une part, pour ce qui est de la communauté issue du choix social, sur la recherche de procédures d'allocation permettant de garantir certaines propriétés d'équité telles que la juste part ou l'absence d'envie, mais sans se préoccuper de considérations de représentation compacte (ce qui est irréaliste), de complexité théorique ou d'algorithmique ;
- ▷ d'autre part, pour ce qui est de la communauté de l'informatique et de l'intelligence artificielle (et plus particulièrement celle des enchères combinatoires), sur l'élicitation et la représentation des préférences et sur le calcul d'un optimum utilitariste, sans aborder le problème de l'équité de la procédure d'allocation.

Ce travail de thèse apporte plusieurs contributions au rapprochement entre le domaine du choix social et le domaine de l'intelligence artificielle et informatique, et à l'étude du problème de partage de manière générale.

Sur le plan de la modélisation, nous avons proposé une extension du modèle *welfariste* cardinal qui intègre le principe de droits exogènes. Si cette extension est fondée sur un ensemble de travaux évoquant ce problème, celui-ci n'avait à notre connaissance jamais été étudié de manière générique.

Sur le plan de la représentation compacte, nous avons effectué une synthèse de l'ensemble des langages dédiés à la représentation des préférences. Nous avons de plus construit deux cadres formels de représentation de problèmes de partage équitable de biens indivisibles, fondés à la fois sur l'ensemble des modèles introduits dans les domaines de la décision individuelle et collective, et sur l'ensemble des travaux sur la représentation compacte de préférences. Ces cadres formels nous ont permis d'introduire un ensemble de résultats de complexité, liés à la recherche ou à l'existence de partages équitables et efficaces dans un contexte où les préférences des agents sont complexes, et par conséquent où les domaines sont combinatoires.

L'étude de la modélisation de l'équité dans le domaine de la micro-économie nous a permis de mettre en évidence un problème algorithmique relativement intéressant et complexe, qui jusqu'ici a été relativement négligé dans la communauté des contraintes : celui de la recherche de solutions leximin-optimales dans un problème combinatoire. Nous avons introduit de nouveaux algorithmes de résolution dédiés à ce problème, et nous avons transposé des algorithmes existant dans d'autres contextes. Cette étude a mené à l'implantation et à la comparaison expérimentale de ces algorithmes, ce qui nous a permis en outre de travailler sur la création d'instances artificielles réalistes de problèmes de partage, et de fournir plusieurs générateurs inspirés de problèmes réels tels que celui du partage de ressources satellitaires.

Un résumé de l'ensemble de ces contributions est présenté sur la figure 6.9.

Perspectives

Bien entendu, les perspectives et les extensions inexplorées de ce travail sont d'autant plus nombreuses que l'étendue du problème de partage de biens indivisibles est grande. Nous proposons quelques pistes qui semblent particulièrement prometteuses.

Un nouveau formalisme de représentation des préférences : Les deux langages de représentation compacte dédiés aux problèmes de partage sur lesquels nous nous sommes concentrés dans ce travail sont fondés sur la représentation logique des préférences des agents. Ce choix est raisonnable : la représentation à base de logique est intuitive, puissante, compacte, et semble relativement appropriée dans le cadre de l'expression de préférences sur des sous-ensembles d'objets. Nous avons cité dans le chapitre 3 un autre langage de représentation de préférences, fondé sur la notion de préférence conditionnelle : les CP-*nets*. Comme nous l'avons fait remarquer, si ce dernier langage est très intuitif et particulièrement intéressant d'un point de vue computationnel, il ne semble pas très approprié à la représentation compacte de préférences dans le domaine du partage (sauf pour un certain type de problèmes très particuliers). Cependant, comme nous l'avons fait remarquer brièvement, une récente extension de ce langage, les TCP-*nets* [Brafman et Domshlak, 2002], introduisent une dimension supplémentaire aux CP-*nets*, en permettant l'expression d'importances relatives et d'importances conditionnelles relatives sur les variables. Cette notion d'importance relative (conditionnelle) exprime simplement une idée simple de hiérarchisation des préférences sur les variables. Cette idée semble particulièrement adaptée aux problèmes de partage, ce qui ouvre éventuellement la voie à un nouveau langage de représentation compacte fondé sur la notion d'importance relative, et dédié à ces problèmes.

L'introduction de compensations monétaires dans le partage : Nous nous sommes concentrés dans ce travail sur le problème de partage de biens indivisibles uniquement. L'introduction de monnaie, ou de toute autre ressource continue — en plus de l'ensemble d'objets — transforme complètement le problème, des définitions les plus basiques (telles que l'absence d'envie) à l'existence de partages satisfaisant ces propriétés. Pour autant, les problématiques abordées dans ce manuscrit ne disparaissent pas : la spécification du problème nécessite toujours un langage de représentation compacte, les problématiques de la complexité et de l'algorithmique subsistent. Le partage de biens indivisibles en présence de compensations est largement étudié dans la communauté des économistes. Il commence à l'être dans celle de l'intelligence artificielle, mais ces travaux se concentrent majoritairement sur une approche distribuée du partage [Endriss *et al.*, 2003; Chevaleyre *et al.*, 2007a; Endriss *et al.*, 2006].

Un rapprochement de l'égalitarisme et de l'absence d'envie : L'égalitarisme et l'absence d'envie sont, comme nous l'avons vu, les deux grandes traductions classiques de la notion subjective d'équité. Elles correspondent à deux visions relativement différentes de cette propriété. D'un côté l'égalitarisme est fondé sur une comparaison interpersonnelle des utilités : cela correspond à la vision d'un arbitre bienveillant, qui partage la ressource selon son propre point de vue de l'inégalité, fondé sur la comparaison absolue des utilités. De l'autre côté, l'absence d'envie est une vision personnelle de l'équité : chaque agent compare sa propre part avec celle des autres et se forge une idée personnelle de l'équité qui ne nécessite pas l'introduction d'outils communs de mesure, tels que des utilités comparables. Cette dualité dans la traduction de la notion d'équité induit bien entendu la question particulièrement intéressante du lien entre ces deux concepts. Nous savons déjà (voir la proposition 4.13) qu'il n'existe pas toujours de partage sans envie parmi les partages optimaux au sens de l'égalitarisme, et qu'il est même très difficile, dans le cas où l'on a affaire à des préférences représentées sous forme compacte, de déterminer si une telle allocation existe. Cependant, de nombreuses questions particulièrement intéressantes restent en suspens. L'égalitarisme favorise-t-il

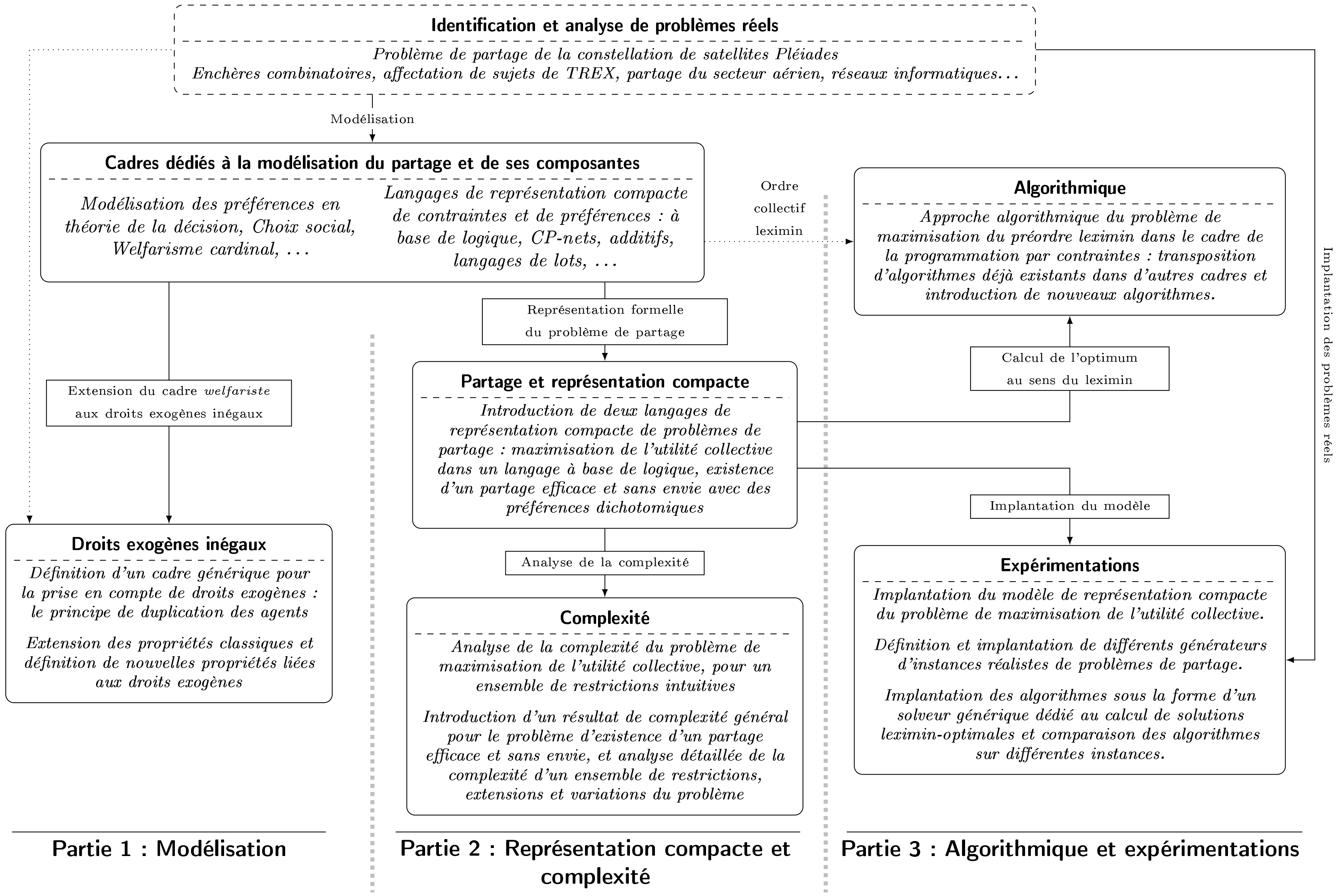


Figure 6.9 — Résumé des contributions de la thèse.

ou défavorise-t-il l'absence d'envie par rapport à un autre critère ? Peut-on mettre en évidence un certain nombre de conditions simples garantissant l'existence d'une allocation égalitariste et sans envie ? Quelle est l'influence de la proximité des préférences des agents sur le lien entre égalitarisme et absence d'envie ? À notre connaissance, ces questions n'ont été que rarement étudiées dans la littérature. On pourra toutefois citer l'exception récente constituée par le travail de [Brams et King, 2005].

Absence d'envie et problèmes à information limitée : Le critère d'absence d'envie part de l'hypothèse selon laquelle chaque agent connaît exactement la part que reçoit chaque autre agent. Cette hypothèse n'est pas vérifiée dans de nombreux problèmes, soit parce que les agents ne connaissent pas ou ont une connaissance partielle des parts des autres agents, soit tout simplement parce que les agents ne se connaissent pas entre eux (voir par exemple les systèmes de commerce en ligne). L'introduction d'information limitée dans les problèmes de partage impose une redéfinition de la notion d'absence d'envie, et donc de l'ensemble des résultats qui lui sont associés. Le problème de l'absence d'envie sous information incomplète peut être abordé sous deux aspects différents. Le premier aspect concerne la connaissance qu'ont les agents des autres agents impliqués dans le partage, ce qui aboutit à une notion d'absence d'envie fondée sur un graphe de connaissance entre les agents : un agent ne peut en envier un autre que s'il existe un arc entre les deux agents. Le deuxième aspect concerne la connaissance qu'ont les agents des parts des autres agents, ce qui aboutit à deux notions de l'envie : envie possible (s'il existe au moins un partage compatible avec les connaissances d'un agent qui mène à l'envie de cet agent pour un autre) ou nécessaire (si tous les partages compatibles avec les connaissances d'un agent aboutissent nécessairement à provoquer l'envie de cet agent). À notre connaissance, l'ensemble de ces problèmes intéressants reste entièrement inexploré, à l'exception d'un travail récent : [Chevaleyre *et al.*, 2007b].

Approximation de l'équité : Les résultats de complexité introduits dans ce travail de thèse, ainsi que les résultats expérimentaux obtenus sur le calcul de solutions leximin-optimales semblent plaider en faveur d'une approche fondée sur l'approximation du critère d'équité (et d'efficacité) utilisé. Cette approche pose deux questions majeures. Tout d'abord se pose le problème de la définition précise de l'approximation du critère, ce qui implique l'introduction d'une mesure de l'erreur commise. Pour un critère fondé sur une fonction d'utilité collective, cette mesure est évidente. Cependant, le cas du préordre leximin pose plus de problèmes : comment définir une bonne approximation de l'optimum dans le cas du leximin ? Pour ce qui est de l'absence d'envie, la question est aussi pertinente. S'agissant d'un critère de décision, cette question se pose en terme de mesure de l'envie, et a été analysée récemment dans quelques travaux [Estivie, 2006; Chevaleyre *et al.*, 2007a; Lipton *et al.*, 2004]. Le deuxième problème crucial posé par l'approximation est algorithmique. La relative inefficacité des algorithmes exacts sur des problèmes de taille élevée, et sur des problèmes concrets de taille raisonnable, montrée dans le chapitre 6, met en évidence la nécessité de développer des algorithmes approchés efficaces, garantissant l'obtention d'une solution correcte dans un temps raisonnable. Le problème se pose en terme de preuve d'existence de schémas d'approximation polynômiaux [Lipton *et al.*, 2004], mais aussi en terme de développement d'algorithmes efficaces. L'étude des différentes méthodes et méta-heuristiques dédiées à la résolution approchée de problèmes d'optimisation combinatoire (recherche locale, algorithmes génétiques, recuit simulé,...) pourra être un bon point de départ.

Stratégies et manipulation : Concluons enfin cette revue des différents problèmes ouverts et des perspectives de cette thèse en abordant le problème de la manipulation des procédures de partage, fondée sur la falsification des préférences par les agents. L'étude de la manipulation, indissociable de tout problème de décision collective, est fondée sur le développement de procédures de choix social ou de mécanismes d'élicitation incitant les agents à révéler leur préférences réelles (*Incentive Compatible Mechanisms*), en rendant la manipulation impossible. Bien entendu, l'«impossibilité» de la manipulation est à prendre dans le sens computationnel du terme : on cherche à rendre cette manipulation le plus difficile possible (concrètement : NP-difficile) pour les agents. Il existe une abondante littérature sur l'étude de la manipulation dans les problèmes de choix social, majoritairement centrée sur le problème du vote [Conitzer *et al.*, 2003]. À notre connaissance, il existe peu de travaux sur cette problématique dans le cas particulier des problèmes de partage de biens indivisibles (notons que ce problème a toutefois été étudié dans le cadre particulier de l'application Pléiades [Lemaître *et al.*, 2002]), ce qui en fait une piste particulièrement intéressante pour la suite de ce travail.

Bibliographie

Ahmet Alkan, Gabrielle Demange et David Gale : Fair allocation of indivisible goods and criteria of justice. *Econometrica*, 59(4):1023–1039, 1991.

Jean-Marc Alliot, Thomas Schiex, Pascal Brisset et Frédérick Garcia : *Intelligence Artificielle et Informatique Théorique*. Cépaduès, 2ème édition, 2002.

Henrik Reif Andersen : An introduction to binary decision diagrams. Rapport technique, Department of Information Technology, Technical University of Denmark, 1999.

Krzysztof R. Apt : *Principles of Constraint Programming*. Cambridge University Press, 2003.

Kenneth J. Arrow, Amartya K. Sen et Kotaro Suzumura, éditeurs. *Handbook of Social Choice and Welfare Volume 1*. Numéro 19 *dans* Handbook in Economics. North-Holland Elsevier, 2002.

Anthony Barnes Atkinson : On the measurement of inequality. *Journal of Economic Theory*, 2:244–263, 1970.

Robert J. Aumann et Sergiu Hart, éditeurs. *Handbook of Game Theory Volume 1*. Numéro 11 *dans* Handbook in Economics. North-Holland Elsevier, 2002.

Fahiem Bacchus et Adam Grove : Graphical models for preference and utility. *Dans* Philippe Besnard et Steve Hanks, éditeurs : *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI-95)*, pages 3–10, Montréal, Canada, août 1995. Morgan Kaufmann.

R. Iris Bahar, Erica A. Frohm, Charles M. Gaona, Gary D. Hachtel, Enrico Macii, Abelardo Pardo et Fabio Somenzi : Algebraic decision diagrams and their applications. *Dans ICCAD-93 : Proceedings of the 1993 IEEE/ACM international conference on Computer-aided design*, pages 188–191, Los Alamitos, CA, USA, 1993. IEEE Computer Society Press.

Michel L. Balinski et H. Peyton Young : *Fair Representation : Meeting the Ideal of One Man One Vote*. Brookings Institution Press, 2ème édition, 2001.

Nicolas Beldiceanu, Mats Carlsson, Sophie Demassey et Thierry Petit : Global constraint catalogue : Past, present and future. *Constraints*, 12(1):21–62, mars 2007.

Nicolas Beldiceanu, Mats Carlsson et Jean-Xavier Rampon : Global constraint catalog. Rapport technique, Swedish Institute of Computer Science, Kista, Sweden, mai 2005.

Chrisitan Bessière, Emmanuel Hebrard, Brahim Hnich et Toby Walsh : The complexity of reasoning with global constraints. *Constraints*, 12(2):239–259, juin 2007.

Christian Bessière et Jean-Charles Régin : Refining the basic constraint propagation algorithm. *Dans* Bernhard Nebel, éditeur : *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, pages 309–315, Seattle, Washington, USA, août 2001. Morgan Kaufmann.

Christian Bessière et Pascal van Hentenryck : To be or not to be... a global constraint. *Dans* Francesca Rossi, éditeur : *Proceedings of the 9th International Conference on Principles and Practice of Constraint Programming (CP-03)*, pages 789–794, Kinsale, County Cork, Ireland, septembre 2003. Springer.

Christian Bessière : Constraint propagation. *Dans* Francesca Rossi, Peter van Beck et Toby Walsh, éditeurs : *Handbook of Constraint Programming*, Foundations of Artificial Intelligence, chapitre 3, pages 29–83. Elsevier, 2006.

Christian Bessière et Marie-Odile Cordier : Arc-consistency and arc-consistency again. *Dans Proceedings of the 11th AAAI Conference on Artificial Intelligence (AAAI-93)*, pages 108–113, Washington, DC, juillet 1993. AAAI Press / MIT Press.

Nicola Bianchessi, Jean-François Cordeau, Jacques Desrosiers, Gilbert Laporte et Vincent Raymond : A heuristic for the multi-satellite, multi-orbit and multi-user management of earth observation satellites. *European Journal of Operational Research*, 177(2):750–762, mars 2007.

Stefano Bistarelli, Hélène Fargier, Ugo Montanari, Francesca Rossi, Thomas Schiex et Gérard Verfaillie : Semiring-based CSPs and valued CSPs : Frameworks, properties, and comparison. *Constraints*, 4(3):199–240, 1999.

Stefano Bistarelli, Ugo Montanari et Francesca Rossi : Constraint solving over semirings. *Dans Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 624–630, Montréal, Canada, août 1995. Morgan Kaufmann.

Stefano Bistarelli, Ugo Montanari et Francesca Rossi : Semiring-based constraint solving and optimization. *Journal of the ACM*, 44(2):201–236, 1997.

Noëlle Bleuzen-Guernalec et Alain Colmerauer : Narrowing a block of sortings in quadratic time. *Dans* Gert Smolka, éditeur : *Proceedings of the 3rd International Conference on Principles and Practice of Constraint Programming (CP-97)*, pages 2–16, Schloss Hagenberg, Austria, octobre 1997. Springer.

Craig Boutilier, Fahiem Bacchus et Ronen I. Brafman : UCP-networks : A directed graphical representation of conditional utilities. *Dans* Jack S. Breese et Daphne Koller, éditeurs : *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI-01)*, pages 56–64, Washington, DC, août 2001. Morgan Kaufmann.

Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos et D. Pool : CP-nets : A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004a.

Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos et David Pool : Preference-based constrained optimization with CP-nets. *Computational Intelligence*, 20(2):137–157, 2004b.

Craig Boutilier et Holger H. Hoos : Solving combinatorial auctions using stochastic local search. *Dans Proceedings of the 17th AAAI Conference on Artificial Intelligence (AAAI-98)*, pages 22–29, Austin, TX, juillet 2000. AAAI Press / MIT Press.

Craig Boutilier et Holger H. Hoos : Bidding languages for combinatorial auctions. *Dans* Bernhard Nebel, éditeur : *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, pages 1211–1217, Seattle, Washington, USA, août 2001. Morgan Kaufmann.

Sylvain Bouveret, Hélène Fargier, Jérôme Lang et Michel Lemaître : Allocation of indivisible goods : a general model and some complexity results. *Dans* Frank Dignum, Virginia Dignum, Sven Koenig, Sarit Kraus, Munindar P. Singh et Michael Wooldridge, éditeurs : *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-05)*, Utrecht, The Netherlands, juillet 2005a. ACM.

Sylvain Bouveret, Hélène Fargier, Jérôme Lang et Michel Lemaître : Un modèle général et des résultats de complexité pour le partage de biens indivisibles. *Dans* Andreas Herzig, Yves Lespérance et Abdel-Illah Mouaddib, éditeurs : *Actes des troisièmes journées francophones Modèles Formels de l'Interaction*. Cépaduès Éditions, mai 2005b.

Sylvain Bouveret et Jérôme Lang : Efficiency and envy-freeness in fair division of indivisible goods : Logical representation and complexity. *Dans* Leslie Pack Kaelbling et Alessandro Saffiotti, éditeurs : *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, Edinburgh, Scotland, août 2005. Professional Book Center.

Denis Bouyssou, Didier Dubois, Marc Pirlot et Henri Prade, éditeurs. *Concepts et Méthodes pour l'Aide à la Décision 2, Risque et incertain*. Lavoisier, 2006.

Denis Bouyssou et Philippe Vincke : Relations binaires et modélisation des préférences. *Dans* Denis Bouyssou, Didier Dubois, Marc Pirlot et Henri Prade, éditeurs : *Concepts et Méthodes pour l'Aide à la Décision*, volume 1, chapitre 2. Lavoisier, 2006.

Ronen I. Brafman et Carmel Domshlak : Introducing variable importance tradeoffs into CP-nets. *Dans* Adnan Darwiche et Nir Friedman, éditeurs : *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 69–76, Edmonton, Alberta, Canada, août 2002. Morgan Kaufmann.

Ronen I. Brafman, Carmel Domshlak et Tanya Kogan : Compact value-function representations for qualitative preferences. *Dans* David Maxwell Chickering et Joseph Y. Halpern, éditeurs : *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI-04)*, pages 51–59, Banff, Canada, juillet 2004. AUAI Press.

Steven J. Brams, Paul H. Edelman et Peter C. Fishburn : Fair division of indivisible items. *Theory and Decision*, 55(2):147–180, 09 2003.

Steven J. Brams, Michal A. Jones et Christian Klamler : Better ways to cut a cake. *Notices of the American Mathematical Society*, 53(11):1314–1321, décembre 2006.

Steven J. Brams et Daniel L. King : Efficient fair division : Help the worst off or avoid envy ? *Rationality and Society*, 17:387–421, 2005.

Steven J. Brams et Alan D. Taylor : *Fair Division — From Cake-cutting to Dispute Resolution*. Cambridge University Press, 1996.

Steven J. Brams et Alan D. Taylor : *The Win-win Solution. Guaranteeing Fair Shares to Everybody*. W. W. Norton & Company, 2000.

Randal E. Bryant : Graph-based algorithms for boolean function manipulation. *IEEE Trans. Computers*, 35(8):677–691, 1986.

Georg Cantor : *Contributions to the Founding of the Theory of Transfinite Numbers*. Dover Publications, New York, 1915.

Li Chen et Pearl Pu : Survey of preference elicitation methods. Rapport technique, École Polytechnique Fédérale de Lausanne, 2004.

Yann Chevaleyre, Paul E. Dunne, Ulle Endriss, Jérôme Lang, Michel Lemaître, Nicolas Maudet, Julian Padget, Steve Phelps, Juan A. Rodríguez-Aguilar et Paolo Sousa : Issues in multiagent resource allocation. *Informatica*, 30:3–31, 2006a.

Yann Chevaleyre, Ulle Endriss, Sylvia Estivie et Nicolas Maudet : Multiagent resource allocation with k-additive utility functions. *Dans Proc. DIMACS-LAMSADE Workshop on Computer Science and Decision Theory*, volume 3 de *Annales du LAMSADE*, pages 83–100, 2004.

Yann Chevaleyre, Ulle Endriss, Sylvia Estivie et Nicolas Maudet : Reaching envy-free states in distributed negotiation settings. *Dans* Manuela M. Veloso, éditeur : *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 1239–1244, Hyderabad, India, janvier 2007a. AAAI Press.

Yann Chevaleyre, Ulle Endriss et Jérôme Lang : Expressive power of weighted propositional formulas for cardinal preference modelling. *Dans* Patrick Doherty, John Mylopoulos et Christopher A. Welty, éditeurs : *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR-06)*, pages 145–152, Lake District, UK, juin 2006b. AAAI Press.

Yann Chevaleyre, Ulle Endriss, Jérôme Lang et Nicolas Maudet : Negotiating over small bundles of resources. *Dans* Frank Dignum, Virginia Dignum, Sven Koenig, Sarit Kraus, Munindar P. Singh et Michael Wooldridge, éditeurs : *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-05)*, pages 296–302, Utrecht, The Netherlands, juillet 2005a. ACM.

Yann Chevaleyre, Ulle Endriss et Nicolas Maudet : On maximal classes of utility functions for efficient one-to-one negotiation. *Dans* Leslie Pack Kaelbling et Alessandro Saffiotti, éditeurs : *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, Edinburgh, Scotland, août 2005b. Professional Book Center.

Yann Chevaleyre, Ulle Endriss et Nicolas Maudet : Allocating goods on a graph to eliminate envy. *Dans Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-07)*, pages 700–705. AAAI Press, juillet 2007b.

Zeev Collin, Rina Dechter et Shmuel Katz : On the feasibility of distributed constraint satisfaction. *Dans* John Mylopoulos et Raymond Reiter, éditeurs : *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 319–324, Sidney, Australia, août 1992. Morgan Kaufmann.

Vincent Conitzer, Jérôme Lang et Tuomas Sandholm : How many candidates are needed to make elections hard to manipulate ? *Dans Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-03)*, pages 201–214, University of Indiana, Indiana, 2003. ACM Press.

Stephen A. Cook : The complexity of theorem-proving procedures. *Dans STOC '71 : Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pages 151–158, New York, NY, USA, 1971. ACM Press.

Martin Cooper et Thomas Schiex : Arc consistency for soft constraints. *Artificial Intelligence*, 154 (1-2):199–227, 2004.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest et Clifford Stein : *Introduction to Algorithms, Second Edition*. MIT Press, 2001.

Sylvie Coste-Marquis, Jérôme Lang, Paolo Liberatore et Pierre Marquis : Expressive power and succinctness of propositional languages for preference representation. *Dans* Didier Dubois, Christopher A. Welty et Mary-Anne Williams, éditeurs : *Proceedings of the 9th International Conference on Principles of Knowledge Representation and Reasoning (KR-04)*, pages 203–212, Whistler, Canada, juin 2004. AAAI Press.

Peter Cramton, Yoav Shoham et Richard Steinberg, éditeurs. *Combinatorial Auctions*. MIT Press, 2006.

Adnan Darwiche et Pierre Marquis : A konwledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, septembre 2002.

Claude d'Aspremont et Louis Gevers : Equity and the informational basis of collective choice. *Review of Economic Studies*, 44(2):199–209, juin 1977.

Claude d'Aspremont et Louis Gevers : Social welfare functionals and interpersonal comparability. *Dans* Kenneth J. Arrow, Amartya K. Sen et Kotaro Suzumura, éditeurs : *Handbook of Social Choice and Welfare Volume 1*, numéro 19 *dans* Handbook in Economics. North Holland Elsevier, 2002.

Simon de Givry, Federico Heras, Matthias Zytnicki et Javier Larrosa : Existential arc consistency : Getting closer to full arc consistency in weighted csps. *Dans* Leslie Pack Kaelbling et Alessandro Saffiotti, éditeurs : *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, Edinburgh, Scotland, août 2005. Professional Book Center.

Simon de Givry, Thomas Schiex et Gérard Verfaillie : Exploiting tree decomposition and soft local consistency in weighted csp. *Dans Proceedings of the 21st AAAI Conference on Artificial Intelligence (AAAI-06)*, Boston, MA, juillet 2006. AAAI Press.

Jean-Paul Delahaye : On se sacrifie... pour nuire aux autres ! *Pour La Science*, dossier spécial 49 : les chemins de la logique, octobre 2005.

Stephen Demko et Theodore P. Hill : Equitable distribution of indivisible items. *Mathematical Social Sciences*, 16:145–158, 1998.

Robert Denda, Albert Banchs et Wolfgang Effelsberg : The fairness challenge in computer networks. *Dans QofIS '00 : Proceedings of the First COST 263 International Workshop on Quality of Future Internet Services*, pages 208–220, London, UK, 2000. Springer-Verlag.

Karine Deschinkel : *Régulation du Trafic aérien par Optimisation Dynamique des Prix du Réseau*. Thèse de doctorat, École Nationale Supérieure de l'Aéronautique et de l'Espace, 2001.

Carmel Domshlak : *Modelling and Reasoning About Preferences with CP-nets*. Thèse de doctorat, Ben-Gurion University, 2002.

Didier Dubois, Hélène Fargier et Henri Prade : Refinements of the maximin approach to decision making in a fuzzy environment. *Fuzzy sets and systems*, 81:103–122, 1996.

Didier Dubois et Philippe Fortemps : Computing improved optimal solutions to max-min flexible constraint satisfaction problems. *European Journal of Operational Research*, 1999.

Didier Dubois, Philippe Fortemps, Marc Pirlot et Henri Prade : Leximin optimality and fuzzy set-theoretic operations. *European journal of operational research*, 130:20–28, 2001.

Paul E. Dunne, Michael Wooldridge et Michael Laurence : The complexity of contract negotiation. *Artificial Intelligence*, 164(1-2):23–46, 2005.

Udo Ebert et Patrick Moyes : Welfare, inequality and the transformation of incomes. the case of weighted income distributions. *Journal of Economics*, Supplement #9:9–50, 2002.

Matthias Ehrgott : *Multicriteria Optimization.* Numéro 491 *dans* Lecture Notes in Economics and Mathematical Systems. Springer, 2000.

Matthias Ehrgott et Xavier Gandibleux, éditeurs. *Multiple Criteria Optimization : State of the Art Annotated Bibliographic Surveys*, volume 52 de *International Series in Operations Research and Management Science.* Springer, 2002.

Ulle Endriss et Nicolas Maudet : On the communication complexity of multilateral tradings. *Dans Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-04)*, New York, NY, juillet 2004. IEEE Computer Society.

Ulle Endriss, Nicolas Maudet, Fariba Sadri et Francesca Toni : On optimal outcomes of negotiations over resources. *Dans Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-03)*, Melbourne, Australia, juillet 2003. ACM.

Ulle Endriss, Nicolas Maudet, Fariba Sadri et Francesca Toni : Negotiating socially optimal allocations of resources. *Journal of Artificial Intelligence Research*, 25:315–348, 2006.

Sylvia Estivie : *Allocation de Ressources Multi-agents : Théorie et Pratique.* Thèse de doctorat, Université Paris Dauphine, décembre 2006.

Boi Faltings : A budget-balanced, incentive-compatible scheme for social choice. *Dans* Peyman Faratin et Juan A. Rodríguez-Aguilar, éditeurs : *Agent-Mediated Electronic Commerce VI*, volume 3435 de *LNAI*, pages 30–43. Springer, 2005.

Boi Faltings : Distributed constraint programming. *Dans* Francesca Rossi, Peter van Beck et Toby Walsh, éditeurs : *Handbook of Constraint Programming*, Foundations of Artificial Intelligence, chapitre 20, pages 699–729. Elsevier, 2006.

Hélène Fargier, Jérôme Lang et Thomas Schiex : Selecting preferred solutions in fuzzy constraint satisfaction problems. *Dans Proceedings of the First European Congress on Fuzzy Intelligent Technologies (EUFIT'93)*, Aachen, 1993.

Hélène Fargier, Jérôme Lang, Michel Lemaître et Gérard Verfaillie : Partage équitable de ressources communes : (1) un modèle général et son application au partage de ressources satellitaires. *Technique et Science Informatiques*, 23(9):1187–1217, 2004a.

Hélène Fargier, Jérôme Lang, Michel Lemaître et Gérard Verfaillie : Partage équitable de ressources communes : (2) éléments de complexité et d'algorithmique. *Technique et Science Informatiques*, 23(9):1219–1238, 2004b.

Hélène Fargier et Pierre Marquis : On valued negation normal form formulas. *Dans* Manuela M. Veloso, éditeur : *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 360–365, Hyderabad, India, janvier 2007. AAAI Press.

Dan S. Felsenthal et Moshé Machover : *The Measurement of Voting Power : Theory and Practice, Problems and Paradoxes*. Edward Elgar Publishing, 1998.

Peter C. Fishburn : *Utility Theory for Decision-Making*. John Wiley & Sons, New York, 1970.

János Fodor et Marc Roubens : *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer Academic Publishers, 1994.

Duncan K. Foley : Resource allocation and the public sector. *Yale Economic Essays*, 7(1):45–98, 1967.

Lester Randolph Ford et Delbert Ray Fulkerson : *Flows in Networks*. Princeton University Press, 1962.

Christian Frei, Boi Faltings et Mounir Hamdi : Resource allocation in communication networks using abstraction and constraint satisfaction. *IEEE Journal on Selected Areas in Communication*, 23 (2):304–320, February 2005.

Eugene C. Freuder : A sufficient condition for backtrack-free search. *Journal of the ACM*, 29 (1):24–32, janvier 1982.

Alan M. Frisch, Brahim Hnich, Zeynep Kiziltan, Ian Miguel et Toby Walsh : Multiset ordering constraints. *Dans* Georg Gottlob et Toby Walsh, éditeurs : *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, Acapulco, Mexico, août 2003. Morgan Kaufmann.

Yuzo Fujishima, Kevin Leyton-Brown et Yoav Shoam : Taming the computational complexity of combinatorial auctions : Optimal and approximate approaches. *Dans* Thomas Dean, éditeur : *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, Stockholm, Sweden, juillet 1999. Morgan Kaufmann.

Lucie Galand et Patrice Perny : Search for compromise solutions in multiobjective state space graphs. *Dans* Gerhard Brewka, Silvia Coradeschi, Anna Perini et Paolo Traverso, éditeurs : *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI-06)*, pages 93–97, Riva del Garda, Italy, août 2006. IOS Press.

Lucie Galand et Patrice Perny : Seach for choquet-optimal paths under uncertainty. *Dans Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence (UAI-07)*, pages 125–132, Vancouver, Canada, juillet 2007. AAAI Press.

Michael R. Garey et David S. Johnson : *Computers and Intractability, a Guide to the Theory of NP-completeness*. Freeman, 1979.

Robert S. Garfinkel et George L. Nemhauser : *Integer Programming*. Wiley, 1972.

Jean-Pascal Gayant : *Risque et décision*. Vuibert, 2001.

Christophe Gonzales et Patrice Perny : GAI networks for utility elicitation. *Dans* Didier Dubois, Christopher A. Welty et Mary-Anne Williams, éditeurs : *Proceedings of the 9th International Conference on Principles of Knowledge Representation and Reasoning (KR-04)*, pages 224–234, Whistler, Canada, juin 2004. AAAI Press.

Christophe Gonzales, Patrice Perny et Sergio Queiroz : Preference aggregation in combinatorial domains using GAI-nets. *Dans* Denis Bouyssou, Fred Roberts et Alexis Tsoukiàs, éditeurs : *Proceedings of the Dimacs-Lamsade workshop on voting theory and preference modelling*, numéro 6 *dans* Annales du LAMSADE, pages 165–179, Paris, France, octobre 2006. CNRS.

Georg Gottlob : Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2:397–425, 1992.

Michel Grabisch : k-order additive discrete fuzzy measure and their representation. *Fuzzy Sets and Systems*, 92:167–189, 1997.

John Charles Harsanyi : Cardinal welfare, individualistic ethics, and interpersonal comparisons of utility. *Journal of political economy*, 63:309–321, 1955.

Joseph Henrich, Robert Boyd, Samuel Bowles, Colin Camerer, Ernst Fehr, Herbert Gintis et Richard : Cooperation, reciprocity and punishment in fifteen small-scale societies. Working Paper 01-01-007, Santa Fe Institute, janvier 2001.

Dorothea K. Herreiner et Clemens Puppe : A simple procedure for finding equitable allocations of indivisible goods. *Social Choice and Welfare*, 19:415–430, 2002.

ILOG : CPLEX 10.0, 2006. `http://www.ilog.com/products/cplex/`.

Philippe Jégou et Cyril Terrioux : Hybrid backtracking bounded by tree-decomposition of constraint networks. *Artificial Intelligence*, 146(1):43–75, 2003.

Frank Jensen, Finn Jensen et Soren Dittmer : From influence diagrams to junction trees. *Dans* Ramon López de Mántaras et David Poole, éditeurs : *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence (UAI-94)*, pages 367–37, Seattle, WA, juillet 1994. Morgan Kaufmann.

Geert M. Jonker, John-Jules Ch. Meyer et Frank P. M. Dignum : A market mechanism for airport traffic planning. *Dans* Carlos Bento, Amílcar Cardoso et Gaël Dias, éditeurs : *Proceedings of the 12th Portuguese Conference on Artificial Intelligence (EPIA-05)*, Lecture Notes in Computer Science, pages 500–511, Covilhã, Portugal, décembre 2005. Springer-Verlag.

Ehud Kalai et Meir Smorodinsky : Other solutions to nash's bargaining problem. *Econometrica*, 43 (3):513–518, 1975.

Richard M. Karp : Reducibility among combinatorial problems. *Dans* R. E. Miller et J. W. Watcher, éditeurs : *Complexity of Computer Computations*, pages 85–103, New York, 1972. Plenum Press.

Ralph L. Keeney et Howard Raiffa : *Decisions with Multiple Objectives : Preferences and Value Tradeoffs*. John Wiley and Sons, 1976.

Donald E. Knuth : *The Art of Computer Programming*, volume 1, fundamental algorithms. Addison-Wesley, 1968.

Serge-Christophe Kolm : *Justice et Équité*. Cepremap, CNRS Paris, 1972.

Arie M. C. A. Koster : *Frequency Assignment - Models and Algorithms*. Thèse de doctorat, Universiteit Maastricht, Maastricht, The Netherlands, 1999.

François Laburthe : CHOCO : Implementing a CP kernel. *Dans Proceedings of TRICS'2000, Workshop on techniques for implementing CP systems*, Singapore, 2000. `http://sourceforge.net/projects/choco`.

Jérôme Lang : Logical preference representation and combinatorial vote. *Annals of Mathematics and Artificial Intelligence*, 42(1):37–71, 2004.

Jérôme Lang : Représentation logique de préférences. *Dans* Denis Bouyssou, Didier Dubois, Marc Pirlot et Henri Prade, éditeurs : *Concepts et Méthodes pour l'Aide à la Décision*, volume 1, chapitre 7. Lavoisier, 2006.

Javier Larrosa et Thomas Schiex : Solving weighted csp by maintaining arc consistency. *Artificial Intelligence*, 159(1-2):1–26, 2004.

Daniel Lehmann, Rudolf Müller et Tuomas W. Sandholm : The winner determination problem. *Dans* Peter Cramton, Yoav Shoham et Richard Steinberg, éditeurs : *Combinatorial auctions*, chapitre 12. MIT Press, 2006.

Daniel Lehmann, Liaden Ita O'Callaghan et Yoav Shoham : Truth revelation in rapid, approximately efficient combinatorial auctions. *Dans Proceedings of the 1st ACM Conference on Electronic Commerce (EC-99)*, Denver, CO, novembre 1999. ACM.

Michel Lemaître, Hélène Fargier et Jérôme Lang : Partage équitable des satellites pléiades-hr. régulation, négociations (lots 1 et 2). Rapport technique RF 1/09401 DCSD, ONERA/DCSD et IRIT/CNRS, octobre 2004.

Michel Lemaître, Gérard Verfaillie et Nicolas Bataille : Exploiting a common property resource under a fairness constraint : a case study. *Dans* Thomas Dean, éditeur : *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 206–211, Stockholm, Sweden, juillet 1999. Morgan Kaufmann.

Michel Lemaître, Gérard Verfaillie, Hélène Fargier et Jérôme Lang : Guidage et partage des satellites Pléiades-HR, étude de protocoles de partage. Rapport technique RF 4/06079 DCSD, ONERA et IRIT, mars 2002.

Kevin Leyton-Brown, Mark Pearson et Yoav Shoham : Towards a universal test suite for combinatorial auction algorithms. *Dans Proceedings of the 2nd ACM Conference on Electronic Commerce (EC-00)*, Minneapolis, MN, octobre 2000. ACM.

Richard Lipton, Evangelos Markakis, Elchanan Mossel et Amin Saberi : On approximately fair allocations of divisible goods. *Dans Proceedings of the 5th ACM Conference on Electronic Commerce (EC-04)*, New York, NY, mai 2004. ACM.

R. Duncan Luce : Semiorders and a theory of utility discrimination. *Econometrica*, 24(2):178–191, avril 1956.

Hanan Luss : On equitable resource allocation problems : a lexicographic minimax approach. *Operations Research*, 47(3):361–378, 1999.

Alan K. Mackworth : Consistency in networks of relations. *Artificial Intelligence*, 8(1):99–118, 1977a.

Alan K. Mackworth : On reading sketch maps. *Dans* Raj Reddy, éditeur : *Proceedings of the 5th International Joint Conference on Artificial Intelligence (IJCAI-77)*, pages 598–606, Cambridge, MA, août 1977b. William Kaufmann.

Jean-Luc Marichal : *Aggregation Operators for Multicriteria Decision Aid.* Thèse de doctorat, Faculté des Sciences de l'Université de Lièges, 1999.

Michael Maschler, Jos A. M. Potters et Stef H. Tijs : The general nucleolus and the reduced game property. *International Journal of Game Theory*, 21:85–106, 1992.

Kurt Mehlhorn et Sven Thiel : Faster algorithms for bound-consistency of the sortedness and the alldifferent constraint. *Dans* Rina Dechter, éditeur : *Proceedings of the 6th International Conference on Principles and Practice of Constraint Programming (CP-00)*, pages 306–319, Singapore, septembre 2000. Springer.

Roger Mohr et Gérald Masini : Good old discrete relaxation. *Dans* Henri Prade, éditeur : *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, pages 651–656, Brighton, England, août 1998. John Wiley and Sons, Chichester.

Ugo Montanari : Network of constraints : Fundamental properties and applications to picture processing. *Inf. Sci.*, 7:95–132, 1974.

Hervé Moulin : *Axioms of Cooperative Decision Making.* Cambridge University Press, 1988.

Hervé Moulin : *Cooperative Microeconomics, A Game-Theoretic Introduction.* Prentice Hall, 1995.

Hervé Moulin : Axiomatic cost and surplus sharing. *Dans* Kenneth J. Arrow, Amartya K. Sen et Kotaro Suzumura, éditeurs : *Handbook of Social Choice and Welfare*, numéro 19 *dans* Handbook in economics, chapitre 6, pages 289–357. North Holland Elsevier, 2002.

Hervé Moulin : *Fair Division and Collective Welfare.* MIT Press, 2003.

John Forbes Nash : The bargaining problem. *Econometrica*, 28:155–162, 1950.

Noam Nisan : Bidding and allocation in combinatorial auctions. *Dans Proceedings of the 2nd ACM Conference on Electronic Commerce (EC-00)*, Minneapolis, MN, octobre 2000. ACM.

Noam Nisan : Bidding languages for combinatorial auctions. *Dans* Peter Cramton, Yoav Shoham et Richard Steinberg, éditeurs : *Combinatorial auctions*, chapitre 9. MIT Press, 2006.

Wlodzimierz Ogryczak : On the lexicographic minimax approach to location problems. *European Journal of Operational Research*, 100:566–585, 1997.

Wlodzimierz Ogryczak et Tomasz Śliwiński : On solving linear programs with the ordered weighted averaging objective. *European Journal of Operational Research*, 148:80–91, 2003.

Meltem Öztürk et Alexis Tsoukiàs : Preference representation with 3-points intervals. *Dans* Gerhard Brewka, Silvia Coradeschi, Anna Perini et Paolo Traverso, éditeurs : *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI-06)*, pages 417–421, Riva del Garda, Italy, août 2006. ACM.

Christos H. Papadimitriou : *Computational Complexity.* Addison–Wesley, 1994.

Judea Pearl : *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. ISBN 0-934613-73-7.

Patrice Perny et Bernard Roy : The use of fuzzy outranking relations in preference modelling. *Fuzzy sets and systems*, 49:33–53, 1992.

Patrice Perny, Olivier Spanjaard et Louis-Xavier Storme : State space search for risk-averse agents. *Dans* Manuela M. Veloso, éditeur : *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2353–2358, Hyderabad, India, janvier 2007. AAAI Press.

Adrian Petcu, Boi Faltings et David Parkes : M-DPOP : Faithful distributed implementation of efficient social choice problems. *Dans* Hideyuki Nakashima, Michael P. Wellman, Gerhard Weiss et Peter Stone, éditeurs : *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-06)*, pages 1397–1404, Hakodate, Japan, mai 2006. ACM.

Marc Pirlot et Philippe Vincke : *Semi Orders*. Kluwer Academic Publishers, Dordrecht, 1997.

M. Pivato : Voting, arbitration and fair division — the mathematics of social choice, novembre 2006. URL : `http://xaravve.trentu.ca/pivato/Teaching/voting.pdf`.

Jos A. M. Potters et Stef H. Tijs : The nucleolus of a matrix game and other nucleoli. *Mathematics of Operations Research*, 17:164–174, 1992.

Cédric Pralet : *A Generic Framework for Representing and Solving Sequential Decision Making Problems with Uncertainties, Feasabilities and Utilities*. Thèse de doctorat, École Nationale de l'Aéronautique et de l'Espace, novembre 2006.

John Rawls : *A Theory of Justice*. Harvard University Press, Cambridge, Mass., 1971. Traduction française disponible aux éditions du Seuil.

Raymond Reiter : A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

Kevin W. S. Roberts : Interpersonal comparability and social choice theory. *Review of Economic Studies*, 47:421–439, 1980.

Jack Robertson et William Webb : *Cake-cutting Algorithms : Be Fair if You Can*. A. K. Peters, Ltd, 1998.

Neil Robertson et Paul D. Seymour : Graph minors. ii. algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986.

Emma Rollón et Javier Larrosa : Bucket elimination for multiobjective optimization. *Journal of Heuristics*, 12(4/5):307–328, septembre 2006.

Jeffrey S. Rosenschein et Gilad Zlotkin : *Rules of Encounter : Designing Conventions for Automated Negotiation Among Computers*. MIT Press, Cambridge, MA, USA, 1994.

Francesca Rossi, Peter van Beck et Toby Walsh, éditeurs. *Handbook of Constraint Programming*. Foundations of Artificial Intelligence. Elsevier, 2006.

Michael H. Rothkopf, Aleksander Pekeč et Ronald M. Harstad : Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.

Bernard Roy : *Méthodologie Multicritère d'Aide à la Décision*. Economica, Paris, 1985.

Tuomas W. Sandholm : Contract types for satisficing task allocation : I. theoretical results. *Dans* Sandip Sen, éditeur : *Proceedings of the AAAI Spring Symposium : Satisficing Models*, pages 68–75, Menlo Park, California, 1998. AAAI Press.

Tuomas W. Sandholm : An algorithm for optimal winner determination in combinatorial auctions. *Dans* Thomas Dean, éditeur : *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 452–547, Stockholm, Sweden, juillet 1999. Morgan Kaufmann.

Tuomas W. Sandholm : Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 134:1–54, 2002.

Tuomas W. Sandholm et Craig Boutilier : Preference elicitation in combinatorial auctions. *Dans* Peter Cramton, Yoav Shoham et Richard Steinberg, éditeurs : *Combinatorial auctions*, chapitre 10. MIT Press, 2006.

Thomas Schiex, Hélène Fargier et Gérard Verfaillie : Valued constraint satisfaction problems : Hard and easy problems. *Dans Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 631–637, Montréal, Canada, août 1995. Morgan Kaufmann.

Thomas Schiex, Hélène Fargier et Gérard Verfaillie : Problèmes de satisfaction de contraintes valués. *Revue d'Intelligence Artificielle*, 9(3):339–373, 1997.

Amartya Sen : *Inequality Reexamined.* Oxford University Press, 1992. Traduction française aux éditions du Seuil.

Amartya K. Sen : *Collective Choice and Social Welfare.* North-Holland, 1970.

Lloyd S. Shapley : A value for n-person games. *Dans* Harold W. Kuhn et Albert W. Tucker, éditeurs : *Contributions to the Theory of Games, vol II*, volume 28 de *Annals of mathematics studies*, pages 307–317. Princeton University Press, Princeton, NJ, 1953.

Barbara M. Smith : Modelling. *Dans* Francesca Rossi, Peter van Beck et Toby Walsh, éditeurs : *Handbook of Constraint Programming*, Foundations of Artificial Intelligence, chapitre 11, pages 377–406. Elsevier, 2006.

Reid G. Smith : The contract net protocol : High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, 1980.

Hugo Steinhaus : The problem of fair division. *Econometrica*, 16(1):101–104, janvier 1948.

Jan Tinbergen : *Redelijke Inkomensverdeling.* N. V. DeGulden Pers., Haarlem, 1953.

Pascal van Hentenryck : *The OPL Optimization Programming Language.* The MIT Press, 1999.

Pascal van Hentenryck, Helmut Simonis et Mehmet Dincbas : Constraint satisfaction using constraint logic programming. *Artificial Intelligence*, 58(1-3):113–159, 1992.

Michel Vasquez et Jin-Kao Hao : A hybrid approach for the 0–1 multidimensional knapsack problem. *Dans* Bernhard Nebel, éditeur : *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, volume 1, pages 328–333. Morgan Kaufmann, août 2001.

Philippe Vincke : Quasi-ordres généralisés et représentation numérique. *Mathématiques et sciences humaines*, 62:35–60, 1978.

Philippe Vincke : *L'aide Multicritère à la Décision.* Ellipses, Bruxelles, 1989.

Georg Henrik Von Wright : *The Logic of Preferences.* Edinburgh University Press, 1963.

Klaus W. Wagner : More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science*, 51:53–80, 1987.

Klaus W. Wagner : Bounded query classes. *SIAM Journal on Computing*, 19(5):833–846, 1990.

Toby Walsh : SAT vs CSP. *Dans* Rina Dechter, éditeur : *Proceedings of the 6th International Conference on Principles and Practice of Constraint Programming (CP-00)*, pages 441–456, Singapore, septembre 2000. Springer.

Ronald R. Yager : On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man, and Cybernetics*, 18:183–190, 1988.

H. Peyton Young : *Equity in Theory and Practice.* Princeton University Press, 1994.

Jiannyang Zhou : A permutation-based approach for solving the job-shop problem. *Constraints*, 2 (2):185–213, 1997.

Daniel John Zizzo et Andrew Oswald : Are people willing to pay to reduce others' incomes ? The Warwick Economics Research Paper Series (TWERPS) 568, University of Warwick, Department of Economics, 2000.

Liste des figures et tableaux

Liste des figures

Liste des tableaux

Liste des symboles

Annexe A
Complexité computationnelle : liste des classes et problèmes

Ce chapitre d'annexe a pour vocation de présenter les classes de complexité utilisées tout au long du document. Nous présentons aussi les problèmes canoniques associés à chaque classe de complexité, ainsi que la liste des problèmes de décision utilisés dans les démonstrations, associés à leur classe de complexité. Nous supposons que le lecteur est familier avec les notions fondamentales de théorie de la complexité : problèmes de décision, langage, machine de Turing, réduction polynomiale. Pour plus d'informations, on pourra se reporter à des ouvrages de référence tels que [Papadimitriou, 1994; Garey et Johnson, 1979].

Pour tout problème de décision $\mathcal{P}$, nous notons $L_{\mathcal{P}}$ un langage «raisonnable» associé à $\mathcal{P}$, et co-$\mathcal{P}$ le problème complémentaire de $\mathcal{P}$. De même, pour une machine de Turing $\mathcal{M}$, on désigne par $L_{\mathcal{M}}$ le langage accepté par $\mathcal{M}$.

A.1 Classes de complexité

Définition A.1 (Classe P) *La classe* P *est formée de l'ensemble des problèmes $\mathcal{P}$ tels qu'il existe une machine de Turing déterministe $\mathcal{M}_D$ s'exécutant en temps polynômial telle que $L_{\mathcal{M}_D} = L_{\mathcal{P}}$.*

Définition A.2 (Classe NP) *La classe* NP *est formée de l'ensemble des problèmes $\mathcal{P}$ tels qu'il existe une machine de Turing non-déterministe $\mathcal{M}_{ND}$ s'exécutant en temps polynômial que $L_{\mathcal{M}_{ND}} = L_{\mathcal{P}}$.*

Le problème NP-complet canonique est le problème [SAT] :

Problème 19: [SAT] [Cook, 1971]

INSTANCE : Une formule booléenne φ en forme normale conjonctive avec 3 littéraux par clause.
QUESTION : φ est-elle satisfiable ?

Définition A.3 (Classe DP) *Un problème $\mathcal{P}$ est dans la classe* DP *si et seulement si il existe un couple de problèmes $(\mathcal{P}_1, \mathcal{P}_2)$ tels que :*
- ▷ $L_{\mathcal{P}} = L_{\mathcal{P}_1} \cap L_{\mathcal{P}_2}$,
- ▷ $\mathcal{P}_1 \in$ NP,
- ▷ $\mathcal{P}_2 \in$ *co*-NP

Remarquons que cette classe n'est *a priori* pas égale à $\mathsf{P} \cap \mathsf{NP}$, à moins que $\mathsf{P} = \mathsf{NP}$. Le problème DP canonique est le problème [SAT-UNSAT]

Problème 20: [SAT-UNSAT]

INSTANCE : Un couple de formules booléennes (φ, φ') en forme normale conjonctive avec 3 littéraux par clause.
QUESTION : Est-il vrai que φ est satisfiable et φ' insatisfiable ?

Les classes de complexité suivantes requièrent la définition de la notion de machine de Turing à oracle :

Définition A.4 (Machine de Turing à oracle) *Étant donné un problème $\mathcal{P}$, une machine de Turing à oracle $\mathcal{P}$ est une machine de Turing multibande $\mathcal{M}^{\mathcal{P}}$, qui possède un état particulier $q_{\mathcal{P}}$ et une bande particulière appelée bande de requête. Lorsqu'elle se trouve dans l'état $q_{\mathcal{P}}$, la machine de Turing passe en un temps unitaire dans un état q_{yes} ou dans un état q_{no}, selon que la chaîne contenue sur la bande de requête appartient à $L_{\mathcal{P}}$ ou pas.*

En d'autres termes, le fonctionnement d'une machine de Turing à oracle est identique à celui d'une machine de Turing normale, excepté le fait qu'elle a la capacité de deviner en un pas de temps la réponse à un problème donné (en faisant appel à un oracle). Pour toute classe de complexité temporelle déterministe ou non-déterministe $\mathbf{C}$, la classe $\mathbf{C}^{\mathcal{P}}$ désigne l'ensemble des problèmes décidés par une machine de Turing du même type et avec la même borne temporelle que pour $\mathbf{C}$, mais avec un oracle de type $\mathcal{P}$.

Si $\mathbf{C}$ est une classe déterministe ou non-déterministe de complexité temporelle polynomiale (telle que P ou NP), $\mathbf{C}^{\mathcal{P}}$ est équivalente à n'importe quelle classe $\mathbf{C}^{\mathcal{P}'}$, pour peu que $\mathcal{P}$ puisse se réduire à $\mathcal{P}'$ en temps polynomial et vice-versa. Ainsi, étant donnée une classe de complexité $\mathbf{C}'$, nous notons $\mathbf{C}^{\mathbf{C}'}$ les classes de type $\mathbf{C}^{\mathcal{P}}$, avec $\mathcal{P} \in \mathbf{C}'$.

Définition A.5 (Hiérarchie polynomiale) *La hiérarchie polynomiale cumulative est l'ensemble des classes de complexité $\mathbf{PH} = \bigcup_{i \in \mathbb{N}} \{\Delta_i, \Sigma_i, \Pi_i\}$ défini récursivement par :*
- ▷ $\Delta_0^p = \Sigma_0^p = \Pi_0^p = \mathsf{P}$
- ▷ $\forall i \geq 0$ *:*
  - $\Delta_{i+1}^p = \mathsf{P}^{\Sigma_i^p}$,
  - $\Sigma_{i+1}^p = \mathsf{NP}^{\Sigma_i^p}$,
  - $\Pi_{i+1}^p = co\text{-}\mathsf{NP}^{\Sigma_i^p}$.

Les problèmes canoniques des classes de la hiérarchie polynomiale s'expriment sous la forme de logique propositionnelle «quantifiée» (à chaque classe de la hiérarchie correspond une séquence précise de quantificateurs) :

Problème 21: [QSAT$_i$]

INSTANCE : Un ensemble de variables booléennes partitionnés en i sous-ensembles $\mathscr{X}_1, \ldots, \mathscr{X}_i$, et une formule propositionnelle $\varphi(\mathscr{X}_1, \ldots, \mathscr{X}_i)$.
QUESTION : Existe-t-il une interprétation des variables de $\mathscr{X}_1$ telle que pour toute interprétation des variables de $\mathscr{X}_2$ il existe une interprétation des variables de $\mathscr{X}_3$, *et caetera* jusqu'à $\mathscr{X}_i$, telle que $\varphi(\mathscr{X}_1, \ldots, \mathscr{X}_i)$ est satisfaite ? En d'autres termes, et par abus de notation, $\exists \mathscr{X}_1 \forall \mathscr{X}_2 \exists \mathscr{X}_3 \ldots Q \mathscr{X}_i \varphi(\mathscr{X}_1, \ldots, \mathscr{X}_i)$ (avec Q désignant $\exists$ si i est pair et $\forall$ si i est impair) est-elle satisfiable ?

Parmi les classes de la hiérarchie polynomiale, les classes du deuxième niveau, Δ_2^{p}, Σ_2^{p} et Π_2^{p}, correspondant respectivement aux classes P^{NP}, $\mathsf{NP}^{\mathsf{NP}}$ et $co\text{-}\mathsf{NP}^{\mathsf{NP}}$ sont de première importance, car

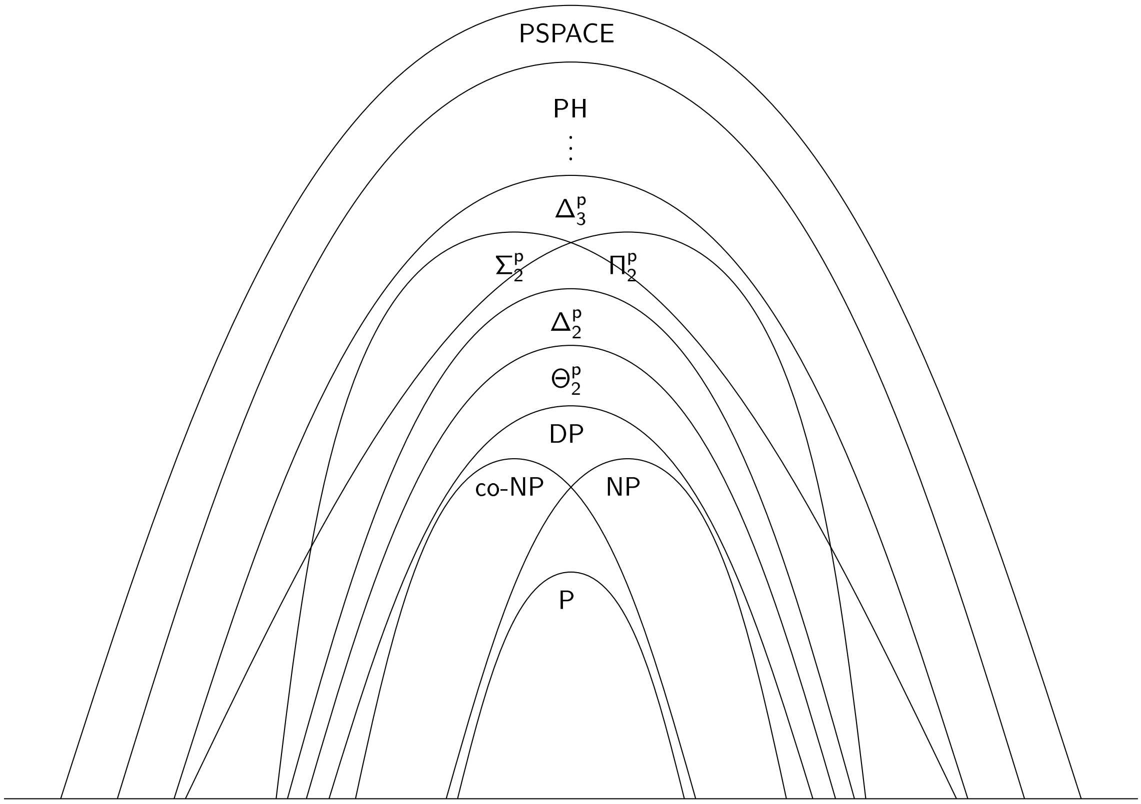


Figure A.1 — Schéma récapitulatif des classes de complexité introduites. Si $\mathsf{P} \neq \mathsf{NP}$, les inclusions de classes sont strictes.

elles correspondent à un ensemble de problèmes issus de l'étude du raisonnement en logique, donc historiquement liés à l'Intelligence Artificielle : raisonnement non-monotone, abduction...

Nous définissons enfin une sous-classe de Δ_2^p, qui correspond aux problèmes dont la résolution nécessite des oracles NP, mais dont le nombre est une fonction logarithmique de la taille des données d'entrée.

Définition A.6 (Classe Θ_2^p) *La classe Θ_2^p est formée de l'ensemble des problèmes décidés par une machine de Turing déterministe $\mathcal{M}^{\mathsf{NP}}$ à oracle de type* NP *qui s'exécute en temps polynômial, et ne nécessitant qu'un nombre d'appels de l'ordre de $O(\log n)$ à l'oracle* NP.

Intuitivement, cette classe correspond aux problèmes impliquant des données numériques, et donc qui peuvent être traités par une recherche dichotomique. Nous pouvons citer comme représentant de cette classe une version du problème [UNWEIGHTED-MAX-SAT] :

Problème 22: [UNWEIGHTED-MAX-SAT]

INSTANCE : Un ensemble $\mathscr{C}$ de clauses et un entier K.
QUESTION : K est-il le nombre maximal de clauses simultanément satisfiables ?

Un résumé de l'ensemble des classes introduites est présenté sur la figure A.1.

A.2 Liste des problèmes introduits

Problème 1 (page 109) : Winner Determination Problem

- ▷ **Instance :** Un ensemble d'agents $\mathscr{N}$, un ensemble d'objets $\mathscr{O}$, un ensemble de fonctions d'utilité $(f_1, \ldots, f_n)$ exprimées sous forme de mises dans un langage d'enchères combinatoires, et un entier K.
- ▷ **Question :** Existe-t-il un partage $\overrightarrow{\pi}$ des objets qui satisfait la contrainte de préemption et tel que $\sum_{i=1}^{n} f_i(\pi_i) \geq K$?

Problème 2 (page 130) : [RSI] (Restricted Skeptical Inference)

- ▷ **Instance :** Un ensemble de formules propositionnelles $\Delta = \{\alpha_1, \ldots, \alpha_n\}$.
- ▷ **Question :** Tous les ensembles maximaux-consitants de Δ contiennent-ils α_1 ?

Problème 3 (page 134) : [SET SPLITTING]

- ▷ **Instance :** Une collection $C = \{\mathscr{C}_1, \ldots, \mathscr{C}_n\}$ de sous-ensembles d'un ensemble fini $\mathscr{S}$.
- ▷ **Question :** Existe-t-il une partition $\langle \mathscr{S}_1, \mathscr{S}_2 \rangle$ de $\mathscr{S}$ telle qu'aucun des sous-ensembles de C n'est entièrement contenu ni dans $\mathscr{S}_1$ ni dans $\mathscr{S}_2$?

Problème 4 (page 140) : [EXACT COVER BY 3-SETS] [Karp, 1972]

- ▷ **Instance :** Un ensemble $\mathscr{S}$ de taille $3q$, et une collection $C = \{\mathscr{C}_1, \ldots, \mathscr{C}_{|C|}\}$ de sous-ensembles à 3 éléments de $\mathscr{S}$
- ▷ **Question :** C contient-il une couverture exacte pour $\mathscr{S}$, c'est-à-dire une sous-collection $C' \subseteq C$ telle que chaque élément de $\mathscr{S}$ apparaît dans exactement un membre de C' ?

Problème 5 (page 144) : [MAX-INDEX-SAT]$_{odd}$[Wagner, 1990]

- ▷ **Instance :** Une suite de formules propositionnelles $(\chi_1, \ldots, \chi_n)$ telle que (χ_i est insatisfiable) $\Rightarrow$ (χ_{i+1} est insatisfiable.)
- ▷ **Question :** L'index maximum i tel que χ_i est satisfiable est-il un nombre impair ?

Problème 6 (page 149) : [MAX-SAT-ASG]$_{even}$ [Wagner, 1987]

- ▷ **Instance :** Une formule propositionnelle χ en forme normale conjonctive, sur un ensemble de variables propositionnelles $V = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$, et une fonction poids w sur les interprétations $v : \mathit{Var} \to \{0, 1\}$, définie par $w(v) \stackrel{\text{def}}{=} \sum_i v(\mathbf{x_i}) \times 2^{i-1}$.
- ▷ **Question :** Est-ce que $\max_{M \text{ modèle de } \chi} w(M)$ est un nombre pair (en d'autres termes la variable $\mathbf{x_1}$ est-elle falsifiée dans le modèle de poids maximal) ?

Problème 7 (page 152) : [PARTITION]

- ▷ **Instance :** Un ensemble fini $\mathscr{S}$ et une taille $s(a) \in \mathbb{N}$ pour tout $a \in \mathscr{S}$.
- ▷ **Question :** Existe-t-il un sous-ensemble $\mathscr{S}' \subseteq \mathscr{S}$ tel que $\sum_{a \in \mathscr{S}'} s(a) = \sum_{a \in \mathscr{S} \setminus \mathscr{S}'} s(a)$?

Problème 8 (page 156) : [MAX-CUF]

- ▷ **Instance :** Une instance $(\mathscr{N}, \mathscr{O}, (\Delta_1, \ldots, \Delta_n), \mathscr{C}, \langle \mathscr{V}_{ind}, \geq_{ind}, \oplus \rangle, \langle \mathscr{V}, \geq, \rangle, g)$ du problème de partage de biens indivisibles et un élément $K \in \mathscr{V}$.
- ▷ **Question :** Existe-t-il une allocation admissible $\overrightarrow{\pi}$ telle que $u_c(\overrightarrow{\pi}) \geq K$?

Problème 9 (page 160) : [INDEPENDENT SET]

- ▷ **Instance :** Un graphe $\mathcal{G} = (V, E)$ et un entier K.
- ▷ **Question :** Existe-t-il un sous-ensemble $\mathscr{S} \subseteq V$ de taille au moins K tel que pour tout couple (v, v') d'éléments de $\mathscr{S}$, $(v, v') \notin E$ (autrement dit, v et v' ne sont pas connectés dans $\mathcal{G}$) ?

Problème 10 (page 161) : [SET PACKING]

- ▷ **Instance :** Une collection C d'ensembles finis, et un entier K.
- ▷ **Question :** Existe-t-il une sous-collection $C' \subseteq C$ d'ensembles disjoints tels que $|C'| \geq K$?

Problème 11 (page 163) : [MAXIMAL MATCHING]

- ▷ **Instance :** Un graphe bipartite pondéré $\mathcal{G}(V = N_1 \cup N_2, E)$, avec $w(v, v')$ désignant le poids d'une arête $\{v, v'\}$, et un entier K.
- ▷ **Question :** Existe-t-il un couplage de poids supérieur ou égal à K, c'est-à-dire un sous-ensemble $E' \subseteq E$ tel que $\sum_{\{v,v'\} \in E'} w(v, v') \geq K$, et $\forall v \in V$, $\nexists (v', v'') \in V^2$ tel que $\{v, v'\} \in E'$ et $\{v, v''\} \in E'$?

Problème 12 (page 164) : [KNAPSACK]

- ▷ **Instance :** Un ensemble fini $\mathscr{S}$ une fonction de valeur $u : \mathscr{S} \to \mathbb{N}$, une fonction de volume $v : \mathscr{S} \to \mathbb{N}$, une capacité maximale v_{max} et un entier K.
- ▷ **Question :** Existe-t-il un sous-ensemble $\mathscr{S}' \subseteq \mathscr{S}$ tel que $\sum_{a \in \mathscr{S}'} v(a) \leq v_{max}$ et $\sum_{a \in \mathscr{S}'} u(a) \geq K$?

Problème 13 (page 164) : [HITTING SET]

- ▷ **Instance :** Une collection C de sous-ensembles $\mathscr{S}_1, \ldots, \mathscr{S}_n$ d'un ensemble $\mathscr{S}$ et un entier $K \leq |\mathscr{S}|$.
- ▷ **Question :** Existe-t-il un sous-ensemble $\mathscr{S}' \subseteq \mathscr{S}$, avec $|\mathscr{S}'| \leq K$, tel que $\mathscr{S}' \cap \mathscr{S}_i \neq \varnothing$ pour tout $\mathscr{S}_i \in C$?

Problème 14 (page 173) : [CSP]

- ▷ **Instance :** Un réseau de contraintes $(\mathscr{X}, \mathcal{D}, \mathscr{C})$.
- ▷ **Question :** $(\mathscr{X}, \mathcal{D}, \mathscr{C})$ a-t-il une solution (une instanciation complète cohérente) ?

Problème 19 (page 249) : [SAT] [Cook, 1971]

- ▷ **Instance :** Une formule booléenne φ en forme normale conjonctive avec 3 littéraux par clause.
- ▷ **Question :** φ est-elle satisfiable ?

Problème 20 (page 250) : [SAT-UNSAT]

- ▷ **Instance :** Un couple de formules booléennes (φ, φ') en forme normale conjonctive avec 3 littéraux par clause.
- ▷ **Question :** Est-il vrai que φ est satisfiable et φ' insatisfiable ?

Problème 21 (page 250) : [QSAT_i]

- ▷ **Instance :** Un ensemble de variables booléennes partitionnés en i sous-ensembles $\mathscr{X}_1, \dots, \mathscr{X}_i$, et une formule propositionnelle $\varphi(\mathscr{X}_1, \dots, \mathscr{X}_i)$.
- ▷ **Question :** Existe-t-il une interprétation des variables de $\mathscr{X}_1$ telle que pour toute interprétation des variables de $\mathscr{X}_2$ il existe une interprétation des variables de $\mathscr{X}_3$, *et caetera* jusqu'à $\mathscr{X}_i$, telle que $\varphi(\mathscr{X}_1, \dots, \mathscr{X}_i)$ est satisfaite ? En d'autres termes, et par abus de notation, $\exists \mathscr{X}_1 \forall \mathscr{X}_2 \exists \mathscr{X}_3 \dots Q \mathscr{X}_i \varphi(\mathscr{X}_1, \dots, \mathscr{X}_i)$ (avec Q désignant $\exists$ si i est pair et $\forall$ si i est impair) est-elle satisfiable ?

Problème 22 (page 251) : [UNWEIGHTED-MAX-SAT]

- ▷ **Instance :** Un ensemble $\mathscr{C}$ de clauses et un entier K.
- ▷ **Question :** K est-il le nombre maximal de clauses simultanément satisfiables ?

Annexe B

Représentation du préordre leximin par une fonction d'agrégation

Cette annexe est dédiée à la représentation du préordre leximin par une fonction d'agrégation, dans le cas où l'ensemble des vecteurs concernés par la fonction d'agrégation est fini. La première section concerne l'évaluation de la taille de l'espace image d'une telle fonction d'agrégation dans un cas particulier réaliste. La deuxième section présente les preuves relatives à la représentation du préordre leximin par des fonctions OWA et somme des puissances, présentées dans le chapitre 1. Enfin, nous abordons dans la troisième section le cas particulier de la famille de fonctions somme des puissances, pour laquelle nous cherchons à calculer l'exposant $p \leq 0$ maximum tel que cette fonction représente le préordre leximin.

B.1 Taille du domaine de la fonction d'agrégation

Considérons une fonction g qui représente le préordre leximin pour un ensemble de profils d'utilité $\mathscr{U}$ fini. Le nombre de classes d'équivalence pour le préordre leximin nous donnera le cardinal de l'ensemble $g(\mathscr{U})$, c'est-à-dire le nombre de valeurs différentes que prendra la fonction g sur l'ensemble des vecteurs de $\mathscr{U}$, ou encore la taille de l'ensemble quotient de $\mathscr{U}$ pour la relation $\sim_{leximin}$.

Nous nous plaçons pour faire ce calcul dans un cas particulier, où nous définissons $\mathscr{U} = \mathscr{V}^n$, avec $\mathscr{V} = [\![1, m]\!]$. Nous avons donc un problème à m^n profils d'utilité possibles.

Proposition B.1 *Soit $g_{leximin} : [\![1, m]\!]^n \to \mathbb{N}$ une fonction représentant le préordre leximin sur $\mathscr{V}^n = [\![1, m]\!]^n$. Alors :*

$$|g_{leximin}([\![1, m]\!]^n)| = \frac{(m + n - 1)!}{(m - 1)!n!}$$

Afin de prouver cette relation, nous avons besoin d'un petit lemme technique :

Lemme 21

$$\forall (m, n) \in \mathbb{N}^{\star 2},\ \frac{(m + n - 1)!}{(m - 1)!} + \frac{(m + n - 2)!}{(m - 2)!} + \cdots + n! = \frac{(m + n)!}{(n + 1)(m - 1)!}. \tag{B.1}$$

Démonstration Ce lemme se prouve facilement par récurrence sur m.

▷ Pour $m = 1$, l'équation (B.1) se réduit à $n! = n!$, qui est trivialement vraie.

▷ Supposons que le lemme soit vrai pour tout n et un certain m. Alors on a :

$$\begin{aligned}\frac{(m+n)!}{m!}+\frac{(m+n-1)!}{(m-1)!}+\cdots+n! &= \frac{(m+n)!}{m!}+\frac{(m+n)!}{(n+1)(m-1)!}\\ &= \frac{(m+n)!\times(n+1)+(m+n)!\times m}{(n+1)(m-1)!\times m}\\ &= \frac{(m+n+1)!}{m!(n+1)}\end{aligned}$$

L'égalité est donc vraie pour $m+1$, ce qui prouve le lemme par récurrence. ▲

Démonstration (Proposition B.1) Nous notons $u_{n,m} = |g_{leximin}([\![1,m]\!]^n)|$. Nous pouvons remarquer que $u_{n,m}$ est exactement égal au nombre de vecteurs ordonnés de $[\![1,m]\!]^n$, et nous avons donc la relation de récurrence sur n suivante : $u_{n,m} = \sum_{k=1}^{m} u_{n-1,k}$. En effet, lorsque l'on ajoute une composante à un vecteur à $n-1$ composantes, il y a $u_{n-1,m}$ vecteurs ordonnés dont la première composante est 1, $u_{n-1,m-1}$ vecteurs ordonnés dont la première composante est 2, etc.

Nous allons prouver la proposition B.1 par récurrence sur n.

▷ Lorsque $n=1$, alors il y a m profils ordonnés différents, et $\frac{(m+n-1)!}{(m-1)!n!} = m$ pour $n=0$, ce qui prouve la proposition pour tout m à $n=1$.

▷ Supposons que pour un m fixé, on ait $u_{n,m} = \frac{(m+n-1)!}{(m-1)!n!}$. Alors d'après la relation de récurrence, on a

$$\begin{aligned}u_{n+1,m} &= u_{n,m}+u_{n,m-1}+\cdots+u_{n,1}\\ &= \frac{(m+n-1)!}{(m-1)!n!}+\frac{(m+n)!}{(m-2)!n!}+\cdots+1\\ &= \frac{1}{n!}\times\left(\frac{(m+n-1)!}{(m-1)!}+\frac{(m+n)!}{(m-2)!}+\cdots+n!\right)\\ &= \frac{(m+n)!}{(n+1)!(m-1)!} \quad \text{(d'après le lemme technique)}\end{aligned}$$

Cela prouve l'égalité pour $n+1$, donc la proposition B.1 par récurrence. ▲

Quel est le comportement de $|g_{leximin}([\![1,m]\!]^n)|$ lorsque m tend vers $+\infty$?

Puisque $|g_{leximin}([\![1,m]\!]^n)| = \dfrac{(m+n-1)\times\cdots\times m}{n!}$, nous avons :

$$\frac{m^n}{n!} \leq |g_{leximin}([\![1,m]\!]^n)| \leq \frac{(m+n-1)^n}{n!}$$

Or, il existe un nombre m_0 tel que $\forall m \geq m_0$, $m \geq n-1$. Donc pour tout $m \geq m_0$, on a :

$$\theta_1 m^n = \frac{m^n}{n!} \leq |g_{leximin}([\![1,m]\!]^n)| \leq \frac{2^n m^n}{n!} = \theta_2 m^n$$

D'où : $\boxed{|g_{leximin}([\![1,m]\!]^n)| = O_{m\to\infty}(m^n)}$

B.2 Représentation du préordre leximin

B.2.1 Par une moyenne pondérée ordonnée

Proposition B.2 *Si l'ensemble des profils d'utilité est fini, alors il existe un OWA qui représente l'ordre leximin.*

Démonstration Soit $\mathscr{U}$ l'ensemble des utilités possibles. On note $m = \min\{|u - u'| \mid (u, u') \in \mathscr{U}^2,\ u \neq u'\}$ et $M = \max\{|u - u'| \mid (u, u') \in \mathscr{U}^2\}$ (ces nombres existent car $\mathscr{U}$ est fini). Puisque $\forall i,\ \frac{x^i - 1}{x^{i+1} - x^i} \sim_{x \to +\infty} \frac{1}{x}$, il est possible de trouver un $K \geq 1$ (assez grand) tel que $\forall i \in [\![1, n]\!],\ \frac{K^i - 1}{K^{i+1} - K^i} < \frac{m}{M}$. On notera g_{lex} l'OWA $g^{(K^n, \ldots, K, 1)}$.

- ▷ Soient $\overrightarrow{u}$ et $\overrightarrow{v}$ deux vecteurs d'utilités tels que $\overrightarrow{u} \succ_{\text{leximin}} \overrightarrow{v}$. Soit j tel que $\forall i < j,\ u_i^\uparrow = v_i^\uparrow$, et $u_j^\uparrow > v_j^\uparrow$. Alors $g_{lex}(\overrightarrow{u}) - g_{lex}(\overrightarrow{v}) = K^{n-j}(u_j^\uparrow - v_j^\uparrow) + \sum_{i=j+1} nK^{n-i}(u_i^\uparrow - v_i^\uparrow)$. Par définition de m et M on a donc :

$$\begin{aligned} g_{lex}(\overrightarrow{u}) - g_{lex}(\overrightarrow{v}) &> mK^{n-j} - M \sum_{i=0}^{j-1} K^{n-i} > mK^{n-j} - M \frac{K^{n-j} - 1}{K - 1} \\ &> mK^{n-j} - M \frac{m}{M} K^j > 0 \end{aligned}$$

  Donc si $\overrightarrow{u} \succ_{\text{leximin}} \overrightarrow{v}$ alors $g_{lex}(\overrightarrow{u}) > g_{lex}(\overrightarrow{v})$.
- ▷ Soient $\overrightarrow{u}$ et $\overrightarrow{v}$ deux vecteurs d'utilités tels que $g_{lex}(\overrightarrow{u}) > g_{lex}(\overrightarrow{v})$. Supposons que $\overrightarrow{v} \succeq_{\text{leximin}} \overrightarrow{u}$. Si $\overrightarrow{v} \succ_{\text{leximin}} \overrightarrow{u}$ alors $g_{lex}(\overrightarrow{v}) > g_{lex}(\overrightarrow{u})$, et si $\overrightarrow{v} \sim_{\text{leximin}} \overrightarrow{u}$ alors $\overrightarrow{v}^\uparrow = \overrightarrow{u}^\uparrow$ donc $g_{lex}(\overrightarrow{u}) = g_{lex}(\overrightarrow{v})$ (par définition d'un OWA). Donc $\overrightarrow{v} \succeq_{\text{leximin}} \overrightarrow{u} \Rightarrow g_{lex}(\overrightarrow{v}) \geq g_{lex}(\overrightarrow{u})$, incompatible avec l'hypothèse de départ. On a donc si $g_{lex}(\overrightarrow{u}) > g_{lex}(\overrightarrow{v})$ alors $\overrightarrow{u} \succ_{\text{leximin}} \overrightarrow{v}$. ▲

B.2.2 Par une fonction somme des puissances

Proposition B.3 *Si l'ensemble des profils d'utilité est fini, alors il existe un $p < 0$ tel que $g^{(p')}$ représente l'ordre leximin pour tout $p' \leq p$.*

Démonstration Soit $\mathscr{U}$ l'ensemble des utilités possibles. On note $m = \min\{|u - u'| \mid (u, u') \in \mathscr{U}^2,\ u \neq u'\}$ et $M = \max\{u \mid u \in \mathscr{U}\}$. Puisque $m/M > 0$, $(1+m/M)^p \xrightarrow[p \to -\infty]{} 0$, donc $\exists p_0 < 0$ tel que $\forall p \leq p_0$, $n(1 + m/M)^p \leq 1$.

- ▷ Soient $\overrightarrow{u}$ et $\overrightarrow{v}$ deux vecteurs d'utilités tels que $\overrightarrow{u} \succ_{\text{leximin}} \overrightarrow{v}$. Soit j tel que $\forall i < j$, $u_i^\uparrow = v_i^\uparrow$, et $u_j^\uparrow > v_j^\uparrow$. Alors $\forall p \leq p_0$, $g^{(p)}(\overrightarrow{u}) - g^{(p)}(\overrightarrow{v}) = \sum_{i=j} n({v_i^\uparrow}^p - {u_i^\uparrow}^p)$. Puisque $u_j^\uparrow > v_j^\uparrow$, nous avons $u_j^\uparrow \geq v_j^\uparrow + m$. De plus, $\forall i \geq j$, $u_i^\uparrow \geq u_j^\uparrow$, donc $u_i^\uparrow \geq v_j^\uparrow + m$, et donc ${u_i^\uparrow}^p \leq (v_j^\uparrow + m)^p$. D'où au final $\sum_{i=j}^n {u_i^\uparrow}^p \leq n(v_j^\uparrow + m)^p$. Puisqu'en outre $v_i \geq 0$ pour tout i, on a donc

$$\sum_{i=j} n({v_i^\uparrow}^p - {u_i^\uparrow}^p) \geq {v_j^\uparrow}^p - n(v_j^\uparrow + m)^p. \tag{B.2}$$

  Or, $v_j^\uparrow \leq M$. Nous avons donc :

$$\frac{m}{v_j^\uparrow} \geq \frac{m}{M} \Rightarrow \left(1 + \frac{m}{v_j^\uparrow}\right)^p \leq \left(1 + \frac{m}{M}\right)^p \Rightarrow n\left(1 + \frac{m}{v_j^\uparrow}\right)^p \leq n\left(1 + \frac{m}{M}\right)^p.$$

Or par définition de p, nous avons $n(1+m/M)^p \leq 1$. D'où $n\left(1+\frac{m}{v_j^\uparrow}\right)^p \leq 1$. Nous avons donc $n\left(v_j^\uparrow + m\right)^p \leq v_j^{\uparrow p}$, et donc $v_j^{\uparrow p} - n\left(v_j^\uparrow + m\right)^p \leq 0$. Cela nous donne au final, avec l'inéquation (B.2), $g^{(p)}(\overrightarrow{u}) - g^{(p)}(\overrightarrow{v}) \geq 0$.

▷ Soit p défini comme ci-dessus et soient $\overrightarrow{u}$ et $\overrightarrow{v}$ deux vecteurs d'utilités tels que $g^{(p)}(\overrightarrow{u}) > g^{(p)}(\overrightarrow{v})$. Supposons que $\overrightarrow{v} \succeq_{\text{leximin}} \overrightarrow{u}$. Si $\overrightarrow{v} \succ_{\text{leximin}} \overrightarrow{u}$ alors $g^{(p)}(\overrightarrow{v}) > g^{(p)}(\overrightarrow{u})$, et si $\overrightarrow{v} \sim_{\text{leximin}} \overrightarrow{u}$ alors $\overrightarrow{v}^\uparrow = \overrightarrow{u}^\uparrow$ donc $g^{(p)}(\overrightarrow{u}) = g^{(p)}(\overrightarrow{v})$ (car $g^{(p)}$ est une fonction symétrique). Donc $\overrightarrow{v} \succeq_{\text{leximin}} \overrightarrow{u} \Rightarrow g^{(p)}(\overrightarrow{v}) \geq g^{(p)}(\overrightarrow{u})$, incompatible avec l'hypothèse de départ. On a donc si $g^{(p)}(\overrightarrow{u}) > g^{(p)}(\overrightarrow{v})$ alors $\overrightarrow{u} \succ_{\text{leximin}} \overrightarrow{v}$. ▲

B.3 Le cas particulier de la fonction somme des puissances

Comme nous l'avons vu précédemment, pour un ensemble fini $\mathscr{U}$ de vecteurs, il existe un $p_0 < 0$ tel que $g^{(p)}$ représente le préordre leximin sur $\mathscr{U}$ pour tout $p \leq p_0$. Nous allons chercher dans cette section à calculer le plus grand p_0 possible pour un $\mathscr{U}$ donné. Afin de clarifier les notations, nous allons noter $q = -p$, et $w^{(q)} = g^{(-p)}$, et nous chercherons donc le plus petit $q > 0$ tel que $w^{(q)}$ représente l'ordre leximin pour un $\mathscr{U}$ donné. Nous supposerons pour simplifier que les vecteurs sont formés de composantes entières pouvant prendre leurs valeurs entre 0 et M.

B.3.1 Le cas critique

Nous avons la proposition suivante :

Proposition B.4 *La fonction $w^{(q)}$ représente l'ordre leximin si et seulement si pour tout $x_0 \in [\![0, M]\!]$, $w^{(q)}(x_0, M, \ldots, M) < w^{(q)}(x_0+1, \ldots, x_0+1)$.*

La preuve de cette proposition est fondée sur un lemme un peu moins fort :

Lemme 22 *La fonction $w^{(q)}$ représente l'ordre leximin si et seulement si pour tout $x_0 \in [\![0, M-1]\!]$, et pour tout $k < n$:*

$$w^{(q)}(\underbrace{x_0, \ldots, x_0}_{k\ fois}, M, \ldots, M) < w^{(q)}(\underbrace{x_0, \ldots, x_0}_{k-1\ fois}, x_0+1, \ldots, x_0+1).$$

Démonstration L'un des sens de l'équivalence est immédiat. Si $w^{(q)}$ représente l'ordre leximin, alors elle doit classer correctement les vecteurs $(\underbrace{x_0, \ldots, x_0}_{k \text{ fois}}, M, \ldots, M)$ et $(\underbrace{x_0, \ldots, x_0}_{k-1\text{fois}}, x_0+1, \ldots, x_0+1)$.

Supposons que pour tous $x_0 \in [\![0, M]\!]$ et $k < n$ on a $w^{(q)}(\underbrace{x_0, \ldots, x_0}_{k \text{ fois}}, M, \ldots, M) < w^{(q)}(\underbrace{x_0, \ldots, x_0}_{k-1 \text{ fois}}, x_0+1, \ldots, x_0+1)$. Alors on a $-(n-k)M^{-q} - x_0^{-q} < -(n-k+1)(x_0+1)^{-q}$. Soient $\overrightarrow{u}$ et $\overrightarrow{v}$ deux vecteurs (que nous supposerons triés dans l'ordre croissant) tels que $\overrightarrow{u} \prec_{\text{leximin}} \overrightarrow{v}$. On note k l'entier tel que $\forall i < k$, $u_i = v_i$ et $u_k < v_k$. Nous avons $-(n-$

$k)M^{-q} - u_k^{-q} < -(n-k+1)(u_k+1)^{-q}$. Donc :

$$
\begin{aligned}
& \sum_{i=1}^{k-1} -u_i^{-q} - (n-k)M^{-q} - u_k^{-q} < \sum_{i=1}^{k-1} -u_i^{-q} - (n-k+1)(u_k+1)^{-q} \\
\Rightarrow \quad & \sum_{i=1}^{k-1} -u_i^{-q} - (n-k)M^{-q} - u_k^{-q} < \sum_{i=1}^{k-1} -v_i^{-q} - (n-k+1)(u_k+1)^{-q} \\
\Rightarrow \quad & \sum_{i=1}^{n} -u_i^{-q} < \sum_{i=1}^{n} -v_i^{-q} \\
\Rightarrow \quad & w^{(q)}(\overrightarrow{u}) < w^{(q)}(\overrightarrow{v}).
\end{aligned}
$$

▲

Démonstration (Proposition B.4) Nous avons :

$$
\begin{aligned}
& w^{(q)}(\underbrace{x_0, \ldots, x_0}_{k \text{ fois}}, M, \ldots, M) < w^{(q)}(\underbrace{x_0, \ldots, x_0}_{k-1 \text{ fois}}, x_0+1, \ldots, x_0+1) \\
\Leftrightarrow \quad & -(n-k)M^{-q} - x_0^{-q} < -(n-k+1)(x_0+1)^{-q} \\
\Leftrightarrow \quad & (n-k)M^{-q} + x_0^{-q} - (n-k+1)(x_0+1)^{-q} > 0
\end{aligned}
$$

Notons $h : (x, n, k, q, M) \mapsto (n-k)M^{-q} + x_0^{-q} - (n-k+1)(x_0+1)^{-q}$. h est dérivable par rapport à k, et on a $\dfrac{\partial h}{\partial k} = -M^{-q} + x_0^{-q}$. Donc pour tout $x_0 < M$, $\dfrac{\partial h}{\partial k} > 0$. Donc $h(x, n, k, q, M) > 0 \Leftrightarrow h(x, n, 1, q, M) > 0$. D'où :

$$
\begin{aligned}
& w^{(q)}(\underbrace{x_0, \ldots, x_0}_{k \text{ fois}}, M, \ldots, M) < w^{(q)}(\underbrace{x_0, \ldots, x_0}_{k-1 \text{ fois}}, x_0+1, \ldots, x_0+1) \\
& \qquad \Leftrightarrow w^{(q)}(x_0, M, \ldots, M) < w^{(q)}(x_0+1, \ldots, x_0+1).
\end{aligned}
$$

▲

B.3.2 Calcul du q minimal

Nous devons trouver le q minimal q_{min} tel que $w^{(q)}(x_0, M, \ldots, M) < w^{(q)}(x_0+1, \ldots, x_0+1)$, c'est-à-dire tel que :

$$
(n-1)M^{-q} + x_0^{-q} - n(x_0+1)^{-q} > 0. \tag{B.3}
$$

Par la suite nous noterons f la fonction suivante :

$$
f : x \mapsto (n-1)M^{-q} + x^{-q} - n(x+1)^{-q}.
$$

B.3.2.1 Tableau de variation de f

Nous avons $f'(x) = nq(x+1)^{-(q+1)} - qx_0^{-(q+1)}$.

$$
\begin{aligned}
f'(x) > 0 & \Leftrightarrow n(x+1)^{-(q+1)} - x_0^{-(q+1)} > 0 \\
& \Leftrightarrow n > \left(\frac{x+1}{x}\right)^{q+1} \\
& \Leftrightarrow \sqrt[q+1]{n} > 1 + \frac{1}{x} \\
& \Leftrightarrow x > \frac{1}{\sqrt[q+1]{n} - 1}
\end{aligned}
$$

La fonction f prend donc son minimum en $\frac{1}{\sqrt[q+1]{n}-1}$, et ce minimum vaut :

$$f(\frac{1}{\sqrt[q+1]{n}-1}) = (n-1)M^{-q} + (\sqrt[q+1]{n}-1)^q(1-n^{\frac{q}{q+1}}).$$

Le tableau de variations de la fonction f est donc le suivant :

x	0		$\frac{1}{\sqrt[q+1]{n}-1}$		$+\infty$
$f'(x)$		$-$	0	$+$	
$f(x)$	$+\infty$	$\searrow$	$f(\frac{1}{\sqrt[q+1]{n}-1})$	$\nearrow$	$(n-1)M^{-q}$

B.3.2.2 Une borne supérieure de q_{min}

Il semble très difficile de résoudre l'équation $f(x) = 0$ dont la solution est nécessaire pour trouver le q minimal cherché (même Maple capitule). En revanche, une condition suffisante pour que f soit positive sur tout $[0, M-1]$ est (1) qu'elle soit décroissante sur tout cet intervalle, et (2) que $f(M-1) > 0$.

Pour que la condition (1) soit vraie, il suffit que $\frac{1}{\sqrt[q+1]{n}-1} > M-1$. On a :

$$\begin{aligned} \frac{1}{\sqrt[q+1]{n}-1} > M-1 &\Leftrightarrow \sqrt[q+1]{n}-1 < \frac{1}{M-1} \\ &\Leftrightarrow \sqrt[q+1]{n} < \frac{M}{M-1} \\ &\Leftrightarrow \frac{ln(n)}{q+1} < ln(\frac{M}{M-1}) \\ &\Leftrightarrow q > \frac{ln(n)}{ln(\frac{M}{M-1})} - 1. \end{aligned}$$

Pour que la condition (2) soit vraie, il faut que :

$$\begin{aligned} f(M-1) > 0 &\Leftrightarrow (n-1)M^{-q} + (M-1)^{-q} - nM^{-q} > 0 \\ &\Leftrightarrow (M-1)^{-q} - M^{-q} > 0 \\ &\Leftrightarrow M > 0. \end{aligned}$$

Cette condition est toujours vraie. Nous avons donc une borne supérieure de q_{min} :

$$\boxed{q_{min} \leq \frac{ln(n)}{ln(\frac{M}{M-1})} - 1}$$

B.3.2.3 Une deuxième borne supérieure de q_{min}

Si q est tel que $f\left(\frac{1}{\sqrt[q+1]{n}-1}\right) > 0$, alors dans f est positive sur tout son domaine de définition. On a :

$$\begin{aligned} f\left(\frac{1}{\sqrt[q+1]{n}-1}\right) > 0 &\Leftrightarrow (n-1)M^{-q} + (n^{\frac{1}{q+1}}-1)^q - n \cdot \left(\frac{n^{\frac{1}{q+1}}-1}{n^{\frac{1}{q+1}}}\right)^q > 0 \\ &\Leftrightarrow (n-1)M^{-q} + (n^{\frac{1}{q+1}}-1)^q \times (1 - \frac{n}{n^{\frac{q}{q+1}}}) > 0 \\ &\Leftrightarrow (n-1)M^{-q} + (n^{\frac{1}{q+1}}-1)^q \times (1 - n^{\frac{1}{q+1}}) > 0 \\ &\Leftrightarrow (n-1)M^{-q} - (n^{\frac{1}{q+1}}-1)^{q+1} > 0 \end{aligned}$$

Cette dernière équation ne semble pas évidente à première vue à résoudre analytiquement. Un angle d'attaque serait peut-être de prouver l'implication suivante :

$$f\left(\frac{1}{\sqrt[q+1]{n}-1}\right) > 0 \Rightarrow \sqrt[q+1]{n} - 1 < \frac{1}{M-1}. \tag{B.4}$$

Dans ce cas, on aurait montré que la valeur minimale de f ne peut pas être positive si la valeur du x pour laquelle $f(x)$ est minimal est avant M. Dans ce cas, le q_{min} serait donc égal à la borne supérieure trouvée précédemment.

Annexe C
Calcul des indices d'inégalité généralisés

Nous détaillons dans cette annexe le calcul des indices d'inégalité généralisés aux droits inégaux. Nous emploierons, dans cette annexe, la notation simplificatrice suivante :

$$u' \stackrel{def}{=} \overrightarrow{u \triangleright^{div}_{\overrightarrow{e}}}.$$

En conséquence, la notation $\overline{u'}$ désignera $\overline{u \triangleright^{div}_{\overrightarrow{e}}}$, c'est-à-dire : $\overline{u'} = \dfrac{1}{m} \displaystyle\sum_{i=1}^{n} u_i = \dfrac{n}{m}\overline{u}$.

C.1 Indices d'Atkinson généralisés

C.1.1 $q \neq 0$

L'indice d'inégalité d'Atkinson classique correspondant à un entier $q \neq 0$ pour n agents s'écrit comme suit :

$$J_q(\overrightarrow{u}) = 1 - \left(\frac{1}{n}\sum_{i=1}^{n}\left(\frac{u_i}{\overline{u}}\right)^q\right)^{\frac{1}{q}}.$$

Appliquons cette définition au profil d'utilités $u' = \overrightarrow{u \triangleright^{div}_{\overrightarrow{e}}}$:

$$J_{q,\overrightarrow{e}}(\overrightarrow{u}) = 1 - \left(\frac{1}{m}\sum_{i=1}^{m}\left(\frac{u'_i}{\overline{u'}}\right)^q\right)^{\frac{1}{q}} = 1 - \left(\frac{1}{m}\sum_{i=1}^{n} e_i\left(\frac{u_i}{e_i\overline{u'}}\right)^q\right)^{\frac{1}{q}} = 1 - \left(\frac{1}{m}\sum_{i=1}^{n} e_i^{1-q}\left(\frac{u_i}{\overline{u'}}\right)^q\right)^{\frac{1}{q}}$$

$$\text{D'où : } \boxed{J_{q,\overrightarrow{e}}(\overrightarrow{u}) = 1 - \frac{m}{n}\left(\frac{1}{m}\sum_{i=1}^{n} e_i^{1-q}\left(\frac{u_i}{\overline{u}}\right)^q\right)^{\frac{1}{q}}}$$

C.1.2 $q = 0$

L'indice d'inégalité d'Atkinson classique correspondant à 0 pour n agents s'écrit comme suit :

$$J_0(\overrightarrow{u}) = 1 - \left(\prod_{i=1}^{n}\frac{u_i}{\overline{u}}\right)^{\frac{1}{n}}.$$

Appliquons cette définition au profil d'utilités $u' = \overrightarrow{u \triangleright_{\overrightarrow{e}}^{div}}$:

$$J_{0,\overrightarrow{e}}(\overrightarrow{u}) = 1 - \left(\prod_{i=1}^{m}\left(\frac{u'_i}{e_i\overline{u'}}\right)^{e_i}\right)^{\frac{1}{m}} = 1 - \left(\prod_{i=1}^{n}\left(\frac{u_i}{e_i\overline{u'}}\right)^{e_i}\right)^{\frac{1}{m}} = 1 - \frac{m}{n}\left(\prod_{i=1}^{n}\left(\frac{u_i}{e_i\overline{u}}\right)^{e_i}\right)^{\frac{1}{m}}$$

$$\text{D'où : } \boxed{J_{0,\overrightarrow{e}}(\overrightarrow{u}) = 1 - \frac{m}{n}\left(\prod_{i=1}^{n}\left(\frac{u_i}{e_i\overline{u}}\right)\right)^{\overline{e_i}}}$$

C.2 Indice de Gini généralisé

C.2.1 Première formulation

La première formulation de l'indice d'inégalité de Gini classique pour n agents s'écrit comme suit :

$$G(\overrightarrow{u}) = \frac{\displaystyle\sum_{k=1}^{n}(k\overline{u} - L(\overrightarrow{u})_k)}{\displaystyle\frac{n}{2}\sum_{i=1}^{n}u_i}.$$

Par simple substitution du profil d'utilités u par le profil $u' = \overrightarrow{u \triangleright_{\overrightarrow{e}}^{div}}$, on obtient l'expression suivante :

$$\boxed{G_{\overrightarrow{e}}(\overrightarrow{u}) = \frac{\displaystyle\sum_{k=1}^{m}(k\overline{u'} - L_{\overrightarrow{e}}(\overrightarrow{u})_k)}{\displaystyle\frac{m}{2}\sum_{i=1}^{n}u_i}}$$

C.2.2 Deuxième formulation

La deuxième formulation de l'indice d'inégalité de Gini classique pour n agents s'écrit comme suit :

$$G(\overrightarrow{u}) = 1 - \frac{1}{n^2\overline{u}}\left(\sum_{k=1}^{n}(2(n-k)+1)u_k^{\uparrow}\right).$$

Appliquons cette définition au profil d'utilités $u' = \overrightarrow{u \triangleright_{\overrightarrow{e}}^{div}}$:

$$G_{\overrightarrow{e}}(\overrightarrow{u}) = 1 - \frac{1}{m^2\overline{u'}}\left(\sum_{k=1}^{m}(2(m-k)+1)u_k'^{\uparrow}\right) = 1 - \frac{1}{nm\overline{u}}\left(\sum_{k=1}^{n}(2m+1)\left(\frac{u_k}{e_k}\right)^{\uparrow\frac{u}{e}} - \underbrace{\sum_{k=1}^{m}2ku_k'^{\uparrow}}_{A}\right)$$

Si nous posons $E_k = \sum_{i=1}^{k} e_i^{\uparrow\frac{u}{e}}$ (avec $E_0 = 1$), nous avons : $A = \displaystyle\sum_{k=1}^{n}\sum_{i=E_{k-1}}^{E_k} 2i\left(\frac{u_k}{e_k}\right)^{\uparrow\frac{u}{e}}$.

En remarquant que $\sum_{i=k_1}^{k_2} i = \frac{(k_2+k_1)((k_2-k_1)+1)}{2}$, nous avons :

$$A = \sum_{k=1}^{n} 2\left(\frac{u_k}{e_k}\right)^{\uparrow\frac{u}{e}} \frac{(E_k+E_{k-1})((E_k-E_{k-1})+1)}{2} = \sum_{k=1}^{n}\left(\frac{u_k}{e_k}\right)^{\uparrow\frac{u}{e}} (2E_{k-1}+e_k^{\uparrow\frac{u}{e}})(e_k^{\uparrow\frac{u}{e}}+1).$$

En remplaçant A dans l'expression de $G_{\overrightarrow{e}}(\overrightarrow{u})$, nous obtenons donc :

$$\boxed{G_{\overrightarrow{e}}(\overrightarrow{u}) = 1 - \frac{1}{nm\overline{u}}\sum_{k=1}^{n}(2m+1-(2E_{k-1}+e_k^{\uparrow\frac{u}{e}})(e_k^{\uparrow\frac{u}{e}}+1))\left(\frac{u_k}{e_k}\right)^{\uparrow\frac{u}{e}}}$$

C.2.3 Troisième formulation

La troisième formulation de l'indice d'inégalité de Gini classique pour n agents s'écrit comme suit :

$$G(\overrightarrow{u}) = \frac{1}{2n^2\overline{u}} \sum_{1\leq i,j\leq n} |u_i - u_j|.$$

Appliquons cette définition au profil d'utilités $u' = \overrightarrow{u \triangleright \frac{div}{\overrightarrow{e}}}$:

$$G_{\overrightarrow{e}}(\overrightarrow{u}) = \frac{1}{2m^2\overline{u'}} \sum_{1\leq i,j\leq m} |u'_i - u'_j| = \frac{1}{2mn\overline{u}} \sum_{1\leq i,j\leq m} |u'_i - u'_j|$$

Les seules valeurs possibles pour $|u'_i - u'_j|$ sont les $|u_k/e_k - u_l/e_l|$, pour $1 \leq k,l \leq n$. Pour $1 \leq i,j \leq m$, u'_i et $u'j$ prennent e_k fois la valeur u_k/e_k, et e_l fois la valeur u_l/e_l. Donc $|u'_i - u'_j|$ prend $e_k e_l$ fois la valeur $|u_k/e_k - u_l/e_l|$ et $e_k e_l$ fois la valeur $|u_l/e_l - u_k/e_k|$.

D'où : $$\boxed{G_{\overrightarrow{e}}(\overrightarrow{u}) = \frac{1}{2nm\overline{u}} \sum_{1\leq k,l\leq m} e_k e_l |\frac{u_k}{e_k} - \frac{u_l}{e_l}|}$$

Index

R

S

T

U

V